

DESARROLLO DE UNA APLICACIÓN CLIENTE-SERVIDOR PARA ANÁLISIS DE EXPERIENCIAS DE USUARIO EN REALIDAD VIRTUAL CON SEGUIMIENTO DE MOVIMIENTO DEL VISOR Y RECONOCIMIENTO GESTUAL DE LAS MANOS

Pablo Díez de la Puente, David Valdivia Martínez y Sheng Zhu

FACULTAD DE INFORMÁTICA
DEPARTAMENTO DE INGENIERÍA DEL SOFTWARE
E INTELIGENCIA ARTIFICIAL
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin Grado en Ingeniería del Software

Madrid, 16 de junio de 2017

Director: Prof. Dr. Federico Peinado Gil
Codirector: Manuel López Ibáñez

Autorización de difusión y utilización

Los alumnos Pablo Díez de la Puente, David Valdivia Martínez y Sheng Zhu, junto al director de este Trabajo de Fin de Grado (TFG), el Prof. Dr. Federico Peinado Gil, y al co-director Manuel López Ibáñez autorizamos a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores tanto la propia memoria de este trabajo, como el código, los contenidos audiovisuales (incluyendo imágenes de los autores), la documentación y el prototipo desarrollado.

Así mismo autorizamos a la UCM a que este trabajo sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Fdo. Pablo Díez de la Puente

Fdo. David Valdivia Martínez

Fdo. Sheng Zhu

Fdo. Prof. Dr. Federico Peinado Gil

Fdo. Manuel López Ibáñez

Agradecimientos

En primer lugar, queremos dar las gracias a todos nuestros familiares y amigos, que nos han ayudado y apoyado durante la realización de este Trabajo Fin de Grado y especial mención a Andrea Watkins Pérez por la revisión de la traducción al inglés.

Agradecemos también al director de nuestro trabajo Federico Peinado Gil y al codirector Manuel López Ibáñez la oportunidad de participar en un proyecto relacionado con uno de los campos más novedosos que existen actualmente, la Realidad Virtual, así como toda su ayuda y dedicación al mismo. También queremos destacar la generosidad de prestarnos el hardware necesario para poder realizar nuestro trabajo.

Ha sido de agradecer el buen trabajo en equipo que ha existido entre los autores y la capacidad de decisión a la hora de resolver ciertos problemas, lo cual ha permitido poder realizar las tareas con mayor facilidad y rapidez. Gracias a todos los que habéis favorecido de una u otra manera a crear este ambiente de trabajo.

Índice general

Índice general	1
Resumen	10
1. Introducción	11
2. Estado de la técnica	16
2.1. Dispositivos de Realidad Virtual	19
2.1.1. Oculus Rift	19
2.1.2. Playstation VR	21
2.1.3. Leap Motion	22
2.2. Entornos de desarrollo para Realidad Virtual	23
2.2.1. Unreal Engine	23
2.2.2. Unity	25
2.3. Videojuegos de Realidad Virtual	27
2.3.1. HellVibe	27
2.3.2. Weightless Remastered	28
2.3.3. Q.U.B.E	29
2.3.4. Pharaoh Puzzle	31
2.3.5. Aboard The Looking Glass	33
2.4. Retos actuales de la Realidad Virtual	35
2.4.1. Retos a nivel tecnológico	35
2.4.2. Retos a nivel de aplicación	36

3. Objetivos y especificación	38
3.1. Objetivos	39
3.2. Especificación de requisitos software	40
3.2.1. Obtención de datos	40
3.2.2. Medidas y variables estadísticas	41
3.2.3. Dinámica del puzzle <i>Cube Board</i>	42
3.2.4. Elementos del escenario	43
3.2.5. Formas de interacción del jugador	44
3.2.6. Tutoriales, menús e interfaces	45
3.2.7. Sistema de secuencias	46
3.2.8. Dinámica del videojuego <i>Escape Room</i>	47
3.2.9. Envío, almacén y visualización de datos	48
4. Metodología y herramientas	49
4.1. Metodología	50
4.2. Plan de trabajo	51
4.3. Herramientas	52
4.3.1. Creación artística	53
4.3.2. Comunicación	54
4.3.3. Alojamiento compartido y control de versiones	55
4.3.4. Realización de la memoria	56
5. <i>Haptic Analytics</i>: Análisis, diseño e implementación	57
5.1. Análisis	58
5.1.1. Diagramas de caso de uso del cliente/servidor	60
5.2. Diseño y arquitectura	62
5.2.1. Diseño de los juegos	62
5.2.2. Diseño de la herramienta	63

5.2.3. Arquitectura	64
5.3. Implementación	66
5.3.1. Lenguajes de programación	66
5.3.2. Tecnologías	67
5.3.3. Desarrollo de los juegos	74
5.3.4. Desarrollo del sitio web	83
6. Pruebas y resultados	90
6.1. Implantación y configuración	90
6.2. Creación de Juego y Experimento en el sitio web	93
6.3. Las fases del juego	97
6.4. Visualización de los resultados	97
7. Conclusiones	101
7.1. Resumen de resultados	102
7.2. Trabajo futuro	104
Aportaciones individuales de los autores	105
Referencias	112
A. Anexo 1. Introduction	113
B. Anexo 2. Conclusions	117
B.1. Summary of result	118
B.2. Future work	120

Resumen

A lo largo de 2016 se ha lanzado al mercado una nueva generación de cascos de Realidad Virtual. Esto ha supuesto un renacer de la inversión y el interés por los dispositivos y aplicaciones relacionados con este campo, desde los sencillos visores para teléfonos móviles, hasta las aparatosas yincanas físicas con visores y controladores hápticos, pasando por los populares vídeos esféricos, en vista omnidireccional de 360 grados.

Las productoras de contenidos digitales buscan aplicar experiencias novedosas de Realidad Virtual en ámbitos como el Entretenimiento, la Publicidad y la Comunicación. Por ello, creadores e investigadores necesitan conocer en profundidad la manera en que los usuarios interactúan con estas aplicaciones. Sin duda, disponer de información objetiva y exhaustiva sobre este comportamiento, permite mejorar el software, potenciando o restringiendo distintos aspectos de la experiencia según su relevancia para los usuarios reales. Cuanto más conocimiento se adquiera durante las pruebas, mayor será la calidad del resultado final.

El objetivo de este proyecto es dar soporte a la investigación del comportamiento real de los usuarios en aplicaciones avanzadas de Realidad Virtual. La propuesta consiste en el desarrollo de un sistema informático, *Haptic Analytics*, que sirve como herramienta de recolección y análisis de datos de la experiencia de usuario en aplicaciones que realizan seguimiento del movimiento del visor y seguimiento de posición y reconocimiento gestual de las manos. Dicho sistema cuenta con una arquitectura cliente-servidor donde el cliente es la propia aplicación objeto de estudio, a la que se le añade un componente adicional, y el servidor es un sitio web con gestión de usuarios y de información, que permite a los investigadores consultar, visualizar y descargar en tiempo real la información de sus experimentos.

Para probar este sistema se han desarrollado dos aplicaciones distintas de Realidad Virtual, diseñadas para el casco Oculus Rift y el reconocedor de manos Leap Motion. La primera consiste en un pequeño puzzle cuya resolución requiere manipular piezas de colores en un tablero. El objetivo es recolocar las piezas según las reglas del juego, usando las manos simplemente a modo de “puntero”. La segunda es un escenario de un videojuego tipo “sala de escape” donde, además de interactuar con objetos de distintas formas, el jugador tendrá que usar gestos manuales para moverse libremente en el entorno virtual.

Las pruebas realizadas con ambas aplicaciones confirman que el sistema *Haptic Analytics* es plenamente funcional, ya que permite recolectar y enviar al sitio web gran cantidad de datos que más tarde pueden ser agregados y convertidos en tablas y gráficas estadísticas de manera automática. Los resultados obtenidos apuntan a que esta puede ser una herramienta muy útil para llevar a cabo futuras investigaciones en el campo de la Realidad Virtual de manera más ágil y eficaz.

Palabras clave

Desarrollo de Software, Desarrollo de Videojuegos, Informática del Entretenimiento, Interacción Persona-Ordenador, Experiencia de Usuario, Realidad Virtual y Aumentada, Tecnología Háptica, Análisis de Datos, Tecnologías Web, Ingeniería del Software, Metodologías Ágiles

DEVELOPMENT OF A CLIENT-SERVER APPLICATION FOR THE
ANALYSIS OF USER EXPERIENCES IN VIRTUAL REALITY WITH VI-
SOR MOVEMENT TRACKING AND HAND SIGNAL RECOGNITION

Abstract

Throughout 2016, a new generation of Virtual Reality helmets such as Oculus Rift, HTC Live and Playstation VR were launched. This has meant a revival of economic investment and public interest in the devices and applications of Virtual Reality, from the simple viewers that are coupled to a mobile phone, to the huge virtual yincanas with viewers and haptic controllers that are realized in Physical installations, passing through the popular spherical videos, recorded with a 360 degree omnidirectional view.

Many technological companies that produce digital content are betting to create innovative immersive experiences that take full advantage of these devices, in fields such as Advertising, Communication or Entertainment. These companies, like research centers, need to know in depth how users use these Virtual Reality applications. It is reasonable to think that having objective and exhaustive information about this behavior of the users will allow to improve the application, enhancing or leaving aside elements as confirmed its real relevance for the experience. In this way, the more knowledge that is acquired in the testing phase with real users, the higher the software product obtained after each iteration will be.

The objective of this project is to give technical support for the investigation on the real user's behavior in advanced applications of Virtual Reality. The proposal consists of the development of a computer system, *Haptic Analytics*, which serves as a tool for collecting and analyzing user experience data in Virtual Reality applications with viewfinder movement tracking and hand gesture recognition. This system has a client-server architecture where the part of the client is integrated into the application itself and the part of the server functions as a website where researchers can view and download the statistical information obtained through their experiments.

To test this system, we have developed Virtual Reality applications, designed for Oculus Rift and Leap Motion. The first one consists on a small puzzle in which you have to manipulate coloured pieces on a board. The objective is to move the pieces according to the game rules, using the hands as a pointer.

The second one is an Escape Room where, in addition to interacting with objects in different ways, the player will have to use manual gestures to move freely through the virtual environment.

The tests performed on both applications confirm that the *Haptic Analytics* system is fully functional, since it allows collecting and sending a great amount of data to the web, where it can be consulted and visualized in a simple and practical way. The results of this work indicate that this can be a very useful tool to streamline future research in the field of Virtual Reality.

Keywords

Software Development, Video Game Development, Entertainment Computing, Human-Computer Interaction, User Experience, Virtual and Augmented Reality, Haptic Technology, Data Analysis, Web Technology, Software Engineering, Agile Methodologies

Capítulo 1

Introducción

El actual auge de la Realidad Virtual (RV) se originó en 2009, cuando un coleccionista aficionado a esta tecnología llamado Palmer Luckey compartió en un foro de Internet su idea de crear el casco de RV más avanzado y económico del mundo. Su proyecto, Oculus Rift¹, obtuvo una financiación asombrosa en 2012 gracias a la plataforma de financiación colectiva Kickstarter², sedujo a veteranos de la industria como John D. Carmack, que se embarcaron en su nueva empresa, y acabó siendo comprado por el gigante Facebook por unos 2000 millones de dólares.

¹Oculus Rift: <https://www.oculus.com/rift/>

²Kickstarter: <https://www.kickstarter.com/>

Los medios de comunicación bautizaron el año 2016 como “el año de la Realidad Virtual” o, en palabras del director de investigación de CSS Insight, “el año en que la conciencia del consumidor de la Realidad Virtual alcanza niveles significativos” (Wood, 2016) ya que coincidieron lanzamientos al mercado de una nueva generación de cascos de RV, como el propio Oculus Rift, HTC Vive³ y Playstation VR⁴. Aunque no se trate de una idea novedosa, como veremos más adelante al estudiar el fenómeno, de alguna manera estos últimos años han supuesto un renacer de la inversión económica y el interés del público por los dispositivos y aplicaciones relacionados con la RV: Facebook con Rift, Valve y HTC con Vive, Sony con su propio visor VR y Microsoft con Hololens, aunque este último está más relacionado con la Realidad Aumentada.

Estos tres dispositivos, que podríamos considerarlos de “gama alta”, son los que mayor repercusión mediática han tenido. Sin embargo también se pueden encontrar dispositivos más económicos, aunque con distintas prestaciones en el mercado, desde los sencillos visores para teléfonos móviles, como GearVR⁵, Cardboard⁶ o Daydream⁷ de Google y muchas otras variantes, hasta las aparatosas yincanas físicas como The Void⁸, con grupos de jugadores reunidos en sus instalaciones portando visores y controladores hápticos, sin olvidarnos de las populares imágenes y vídeos esféricos, o inmersivos, tomados utilizando para ello una vista omnidireccional de 360 grados.

³HTC Vive: <https://www.vive.com/eu/>

⁴PlayStation VR: <https://www.playstation.com/es-es/explore/playstation-vr/>

⁵Gear VR: www.samsung.com/GearVR

⁶Google Cardboard: <https://vr.google.com/cardboard/>

⁷Google Daydream: <https://vr.google.com/daydream/>

⁸The Void: <https://www.thevoid.com/>

Los estudios de desarrollo y las empresas editoras de contenidos digitales buscan ofrecer experiencias novedosas de RV en ámbitos como el Entretenimiento, la Publicidad y la Comunicación. Particularmente la tecnología se ha propuesto llegar a los usuarios principalmente a través de los videojuegos, con títulos que ofrecen experiencias sensoriales llamativas como *The Climb* (Crytek, 2016), *EVE Valkyrie* (Cooper, 2016) o *Superhot VR* (SuperHOT team, 2016). En los videojuegos no hay una exigencia de precisión ni un riesgo tan grande para la salud o los negocios como lo puede haber en aplicaciones de propósitos alejados del ocio y la diversión.

Para innovar y sacar provecho de la tecnología, tanto los creadores como los investigadores necesitan conocer en profundidad la manera en que los usuarios interactúan con estas aplicaciones, la forma en que los usuarios disfrutan de sus juegos o se benefician del uso de sus aplicaciones. Sin duda, disponer de información objetiva y exhaustiva sobre este comportamiento, permite mejorar el software, potenciando o restringiendo distintos aspectos de la experiencia según se valore su relevancia para los usuarios reales. Por ejemplo, conociendo qué niveles, personajes u objetos de un videojuego han gustado más o menos a los jugadores para poder crear títulos más entretenidos, impactantes e incluso adaptados al gusto de estos. Como premisa de este trabajo, consideramos que cuanto más conocimiento se adquiera durante las pruebas, realizadas en cada iteración en el caso de un proceso de desarrollo iterativo, más satisfactoria podrá ser la calidad del resultado final, del producto software obtenido.

La motivación de este trabajo surge al tomar conciencia de esta necesidad, que no sólo es una demanda de la industria sino que resulta evidente también desde el punto de vista de un grupo de investigación como en el que trabaja nuestro director, donde se investiga en videojuegos narrativos y nuevas tecnologías como la RV. Teniendo en cuenta todo esto, la herramienta que nos hemos propuesto desarrollar es un sistema informático que dé soporte a la investigación del comportamiento real de los usuarios en aplicaciones avanzadas de RV, concretamente juegos o aplicaciones que realizan tanto seguimiento del movimiento del visor como seguimiento de posición y reconocimiento de gestos hechos con las manos.

El sistema propuesto, *Haptic Analytics*, se encargará en primer lugar de recoger información sobre el usuario (cómo se desenvuelve por el juego, hacia donde dirige su atención, con qué elementos interactúa y de qué manera). Para poder realizar esta primera recogida de datos, crearemos un componente adicional del sistema que se pueda integrar en la propia aplicación que se va a estudiar, así como un par de aplicaciones de ejemplo que lleven este complemento ya integrado y hagan uso de él, para que sirvan como referencia al desarrollador de la aplicación.

En segundo lugar, el sistema deberá ofrecer a los investigadores, una vez realizados los experimentos con usuarios reales, un sitio web al que poder acceder, previa identificación, y consultar, visualizar y descargar cómodamente desde allí toda aquella información estadística que deseen: tablas y gráficas con datos agregados según diferentes criterios, para su posterior tratamiento y análisis.

Tras haber intentado explicar el contexto de este trabajo en el presente capítulo, exponemos a continuación cual es la estructura del resto de la memoria. El Capítulo 2 revisa el estado de la técnica en RV, los dispositivos, entornos de desarrollo y herramientas existentes, así como algunos videojuegos que hemos tomado como referencia para el desarrollo de nuestras aplicaciones de ejemplo. En el Capítulo 3 se detallan los objetivos de este proyecto, elaborando una especificación de requisitos que servirá como base para el posterior desarrollo. El plan de trabajo general y las herramientas relacionadas con la comunicación y la organización de trabajo que han sido utilizadas se presentan en el Capítulo 4, así como las pequeñas influencias de distintas metodologías. El Capítulo 5 contiene el grueso de la contribución de este trabajo, ya que es donde se detalla el análisis, el diseño e incluso la implementación del sistema propuesto, *Haptic Analytics*. También se incluye en este capítulo todo lo relativo al desarrollo de las dos aplicaciones de ejemplo, que ilustran el uso de este sistema. Las pruebas realizadas sobre dichas aplicaciones y los resultados obtenidos se discuten en el Capítulo 6. Finalmente, el Capítulo 7 recoge nuestras conclusiones sobre este proyecto, cuyos resultados han sido favorables y que ha generado varias líneas de trabajo futuro interesantes para seguir mejorando la herramienta propuesta.

Capítulo 2

Estado de la técnica

Los primeros intentos de construir un dispositivo de Realidad Virtual (RV) se vieron plasmados en el artículo pionero *The Ultimate Display* (Sutherland, 1965). Años atrás, algunos científicos habían comenzado a idear los principios fundamentales de esta tecnología, como Charles Wheatstone en el año 1844 con la creación del estereoscopio (Figura 2.1). Este fue el primer objeto utilizado con la intención de experimentar una realidad alternativa a través del sentido de la vista.



Figura 2.1: Estereoscopio de Charles Wheatstone.

Su función principal era la misma que la de un dispositivo de RV actual: presentar dos imágenes iguales a cierta distancia, que serán observadas por cada ojo de manera independiente pero que el cerebro combinará en una única “imagen tridimensional”.

Casi 120 años después de la creación del estereoscopio, en 1961, se construye el primer casco de RV. Este, a diferencia del estereoscopio de Wheatstone, permitía ver imágenes en movimiento, además de estar equipado con un sensor magnético que ayudaba a conocer la orientación de la cabeza del usuario. Esta característica, el seguimiento de la posición y orientación de la cabeza, sigue siendo, a día de hoy, fundamental para cualquier sistema de RV moderno.

La primera frase que describe lo que puede ser la RV se encuentra en el artículo anteriormente citado, en el que leemos: “La pantalla es una ventana a través de la cual uno ve un mundo virtual. El desafío es hacer que ese mundo se vea real, actúe real, suene real, se sienta real” (Sutherland, 1965). Este pensamiento fue probablemente el que llevó a Ivan Sutherland a crear el primer casco de RV utilizando tubos de rayos catódicos y un sistema mecánico que hacía posible el seguimiento del usuario. Posteriormente, en 1968, Sutherland inventó junto con David Evans el primer “generador” de escenarios con imágenes tridimensionales y datos almacenados.

Como la mayoría de las nuevas tecnologías, esta también tuvo una gran aplicación en el ámbito militar, concretamente en la simulación de vuelos, como es el caso del sistema “Cabina Virtual” (Furness, 1981). Sin embargo, su mayor aportación llega un año después con el simulador más avanzado en la época, denominado VCASS (Visually Coupled Airborne Systems Simulator).

Aunque lo primero que uno relaciona con la RV son los dispositivos para la cabeza, gafas o visores llamados HMD (del inglés *Head-Mounted Display*), también hay otros componentes hardware que permiten disfrutar de ella, como los guantes u otros sensores de captación del gestos de las manos.

El primer invento patentado de este estilo fue el guante háptico de Gary Grimes, el cual en 1983 era capaz de reconocer la posición y orientación de la mano. Este aparato aspiraba a sustituir el teclado de un ordenador por el reconocimiento de gestos realizados con los dedos.

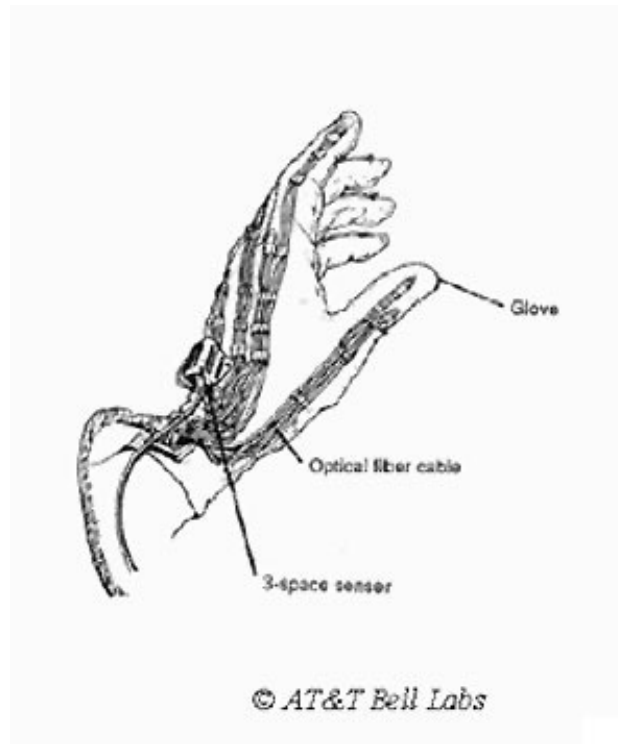


Figura 2.2: Guante háptico de Gray Grimes.

El sistema asociado a este guante permitía reconocer las posiciones de la mano con precisión suficiente para producir caracteres alfanuméricos como respuesta. Se componía por sensores de flexión en los dedos, sensores táctiles en las yemas y sensores de orientación y posición en la muñeca.

Otro dispositivo háptico novedoso es Airborne Ultrasound Tactile Display (AUTD), que recuerda a Leap Motion, el dispositivo que se utiliza en este proyecto como explicaremos más tarde. Fue desarrollado por un equipo de investigadores del departamento de Física y Computación de la Universidad de Tokio y difiere del anterior en que el usuario no necesita vestir guantes. El sistema fue presentado en 2008, es mucho más reciente, y su función es generar sensaciones táctiles vinculadas a imágenes virtualizadas mediante ultrasonidos, permitiendo al usuario “tocar” el mundo virtual simplemente moviendo la mano. La idea que llevó a crear este dispositivo fue que, si el sonido puede hacer mover nuestro tímpano con sus ondas, también puede hacer mover otros objetos. Por ello se pretende utilizar varios emisores sincronizados, de manera que se puedan enfocar las ondas a un punto concreto. Si ese punto está situado en la mano, sentiremos que lo “tocamos” aunque no lo veamos.

En este capítulo vamos a revisar los dispositivos cuya utilización ha sido considerada en este proyecto, así como los entornos de desarrollo de videojuegos disponibles, y algunos juegos de RV que hemos tomado de referencia para crear las aplicaciones de prueba del Capítulo 5.

2.1. Dispositivos de Realidad Virtual

En esta sección nos centramos en los dispositivos hardware que debemos conocer para la realización de este proyecto.

2.1.1. Oculus Rift

Son las gafas de RV desarrolladas por la empresa Oculus VR, cuyo primer *kit* para desarrolladores, el DK1, fue lanzado el 29 de marzo de 2013 y cuya versión para consumidores fue lanzada el 28 de marzo de 2016.

Desde la campaña de financiación colectiva que puso en marcha el proyecto hasta su lanzamiento final orientado al consumidor, este dispositivo ha pasado por las siguientes versiones.

- **Prototipo inicial.** Primer prototipo creado por Luckey. El dispositivo tenía una pantalla LCD de 5.6 pulgadas con un campo de visión de 90 grados en horizontal y 110 grados vertical.
- **Development Kit 1.** Primera versión dedicada al desarrollo lanzado el 29 de marzo de 2013 que cuenta con una pantalla LCD de 7 pulgadas, campo de visión diagonal de 110 pulgadas y una resolución total de 1280 x 800 píxeles.
- **Prototipo HD.** Prototipo que hace uso de una pantalla LCD de 1080p presentado en el E3¹.
- **Prototipo *Crystal Cove*.** Prototipo que hace uso de una pantalla OLED y también mejora el sistema de rastreo. Fue presentado en el CES² de 2014.
- **Development Kit 2.** Esta versión proviene del prototipo *Crystal Cove*, mejorando varios aspectos frente al DK2 como por ejemplo una mayor resolución (1080p), pantalla OLED, mayor tasa de refresco y rastreo de posición u omisión de la caja de controlador externo. Fue lanzada en julio de 2014.
- **Prototipo *Crescent Bay*.** Este prototipo, presentado en septiembre de 2014, mejora el DK2 con una mayor resolución, menor peso, audio integrado y rastreo de orientación de la cabeza 360 grados.
- **Versión de consumidor.** Tras varias versiones, Oculus presenta Oculus Rift para los consumidores, la llamada CV1, en mayo de 2015 y se lanza oficialmente en marzo de 2016. Esta versión es una mejora del prototipo *Crescent Bay* con una tasa de refresco de 90 Herzios por ojo, con mejoras estéticas y ergonómicas del caso.

¹Esta convención es la más importante de la industria de los videojuegos y se celebra cada año en torno a junio o julio. Se exponen nuevas novedades en software y hardware junto a una entrega de premios a los mejores juegos la convención.

²Consumer Electronics Show (CES) o Feria de Electrónica de Consumo, un evento anual celebrado en enero en la ciudad de Las Vegas. En esta feria se enseñan nuevos productos de electrónica que aparecerán pronto para los consumidores, y se cierran contratos entre empresas del sector.



Figura 2.3: Gafas Oculus Rift Versión Development Kit 2.

2.1.2. Playstation VR

Playstation VR, conocido como Project Morpheus en su fase de desarrollo, es el dispositivo de RV desarrollado por Sony Interactive Entertainment ³ y lanzado al público el 13 de octubre de 2016.

Está diseñado para ser compatible con la consola Playstation 4 ⁴.

Este dispositivo cuenta con una pantalla OLED de 5.7 pulgadas con una resolución de hasta 1080p (1920 x 1080) para cada ojo y capaz de mostrar contenido a 120 *frame* por segundo. Además cuenta con sonido 3D y micrófono para una experiencia más inmersiva.

³Sony Interactive Entertainment: <https://www.playstation.com>

⁴PS4: <https://www.playstation.com/es-es/explore/ps4/>

En marzo de 2016, Sony anunció que había 130 desarrolladores trabajando en contenidos para Playstation VR, siendo un gran atractivo para hacer frente a otros dispositivos VR que llevan más tiempo en el mercado.

A febrero de 2017, se ha vendido 915.000 unidades de Playstation VR, superando todas las expectativas de la compañía.

2.1.3. Leap Motion

Leap Motion es un dispositivo sensor de manos y dedos que hace uso de dos cámaras infrarrojas monocromáticas para capturar los movimientos de las manos, movimientos que son almacenados en forma de matriz y enviados por un cable USB hacia el ordenador donde se procesan. Allí los datos son analizados con el software de Leap Motion que hace uso de un algoritmo propietario y secreto de la compañía. El sensor estaba diseñado para usarse encima de una superficie, pero con el auge de los visores de RV, ahora es posible montarlo sobre estas, ya sean Oculus Rift o HTC Vive.



Figura 2.4: Dispositivo Leap Motion usándose en el modo de sobremesa.

Leap Motion Orion es el software que mejora las funcionalidades básicas del Leap Motion cuando se combina con Oculus Rift. Esta actualización ha mejorado considerablemente las capacidades de seguimiento de las manos permitiendo reconocerlas ahora incluso en ángulos extremos y de mayor dificultad. Además de aumentar el rango de reconocimiento, ahora el software permite que el usuario estire más que antes los brazos y que de esta manera pueda ver la representación de sus manos en la realidad virtual.

Por otro lado, lo que más ha cambiado en la versión del software de Leap Motion en sobremesa es la capacidad de reconocer acciones de agarrar y pellizcar junto a la función principal de este dispositivo, el reconocimiento de las manos.

2.2. Entornos de desarrollo para Realidad Virtual

Tras hablar con el director y el coordinador de este trabajo y buscar información acerca del desarrollo de aplicaciones en RV, decidimos crear nuestra herramienta y los juegos donde ilustrar su funcionamiento mediante el entorno de desarrollo Unity⁵. Además de saber que Unity ofrece muchísima información, tutoriales y ayudas para el desarrollador, también es una tecnología que tiene requisitos técnicos más bajos para trabajar con ella que otros entornos como Unreal Engine ⁶, aunque ciertamente los requisitos para la RV son siempre más altos de lo normal. Otro motivo para escoger Unity en vez de Unreal Engine es que los miembros del proyecto ya habíamos trabajado previamente haciendo alguna pequeña aplicación con esta herramienta y por lo tanto nos sentíamos más cómodos desarrollando con ella en vez de con cualquier otra.

2.2.1. Unreal Engine

Actualmente en el mercado existen varias opciones de entornos de desarrollo con un potencial increíble para la Realidad Virtual, capaces de procesar y renderizar elementos en tres dimensiones con un gran realismo (véase figura 2.5).

⁵Unity: <https://unity3d.com/es>

⁶Unreal: <https://www.unrealengine.com>

Muchos de estos entornos de desarrollo son propietarios y pertenecen a empresas que desarrollan sus propios juegos con ellos, como puede ser el caso de Frostbite Engine ⁷ que utilizan los desarrolladores de *EA Digital Illusions Creative Entertainment* (más conocidos como DICE), con el que se han creado grandes títulos de entretenimiento como son la mayor parte de la saga *Battlefield* (2002) y el popular *FIFA 17* (2016).

Sin embargo, algunas de estas compañías deciden licenciar su producto para que grandes o pequeñas empresas de desarrollo independiente puedan utilizar sus herramientas y no tengan que dedicar el enorme esfuerzo que requiere el desarrollo de un motor propio.

Un caso concreto, con el que nos planteamos crear nuestro proyecto, fue Unreal Engine. Este entorno creado y desarrollado por Epic Games se encuentra actualmente en su cuarta versión, que se licencia con un modelo de ingresos basado exclusivamente en porcentaje de beneficios, a diferencia de sus predecesoras. Unreal Engine destaca por su gran potencia de renderización, así como un complejo sistema de materiales, iluminación, físicas y otras características que consiguen un resultado de un realismo superior. Además esta herramienta permite crear videojuegos multiplataforma, al igual que Unity, incluyendo móviles y, por supuesto, dispositivos de RV. Algunos de los juegos desarrollados con el motor de Unreal Engine son *Depth VR* (Digital Confectioners, 2017), *Batman Arkham VR* (Rocksteady Studios, 2016) o *Tekken 7* (Namco Bandai Games, 2015).

De todas maneras, obtener un resultado espectacular en Unreal Engine lleva por detrás mucho tiempo y esfuerzo, con equipos muy grandes de desarrolladores. Precisamente porque no disponemos de semejantes recursos, y por nuestra inexperiencia en la creación de contenido en Unreal Engine, hemos descartado esta opción para este proyecto en favor de Unity.

⁷Frostbite Engine: <https://www.ea.com/frostbite>



Figura 2.5: A Boy and his Kite, demo que presentaron con Unreal Engine 4.8.

2.2.2. Unity

Unity, el entorno de desarrollo de videojuegos multiplataforma creado por Unity Technologies tiene como una de las características principales dar soporte a la compilación sobre diferentes tipos de plataforma, desde la Web al ordenador personal, a los dispositivos móviles, consolas, en televisores inteligentes (Smart TV) y, por supuesto, las plataformas que más nos interesan aquí, las de RV.

El motor de Unity posee características similares a la de otros potentes motores de videojuegos en cuanto a la renderización de complejos objetos en tres dimensiones, ofreciendo sistemas de iluminación, materiales y físicas, entre otros, para alcanzar un mayor realismo. Sin embargo lo que más ha diferenciado a Unity de las demás opciones ha sido el permitir a desarrolladores independientes utilizar una herramienta sin necesidad de adquirir licencias excesivamente cuantiosas. La compañía se ha focalizado en “democratizar” el desarrollo de videojuegos y ofrecer una herramienta integrada y sencilla que permite crear juegos tanto en dos como en tres dimensiones, casi a cualquier persona.

Otra ventaja que tiene Unity y que a su vez lo vuelve a diferenciar de otras herramientas es que tiene una tienda interna muy bien surtida, denominada Unity Asset Store, a la que pueden acceder todos los desarrolladores tanto para descargar contenido como para publicarlo. El contenido puede ir desde herramientas que facilitan tareas hasta paquetes de modelos en dos y tres dimensiones, texturas, sonidos o animaciones. Podemos encontrar recursos tanto gratis como de pago. Algunos juegos conocidos que usan el motor de unity, son *Hearthstone: Heroes of Warcraft* (Blizzard Entertainment, 2014), *Pokemon GO* (Niantic Inc, 2016), *Nihilumbra* (BeautiFun Games, 2012) o *Pillars Of Eternity* (Obsidian Entertainment, 2015), lo que refleja de todo lo que es capaz este entorno de desarrollo.

Sin embargo Unity trabaja también cuenta con distintos servicios, tanto para juegos desarrollados en realidad virtual como para juegos desarrollados en cualquier otra plataforma. Uno de ellos es Analytics que permite hacer un análisis básico e incorporado para el desarrollo de un juego, permitiendo ayudar a mejorar la economía dentro del juego y la experiencia para el jugador. Este contiene un monitor de métricas donde obtener un resumen de alto nivel de cómo se está usando el juego, un explorador de datos, el cual profundiza más en los casos de usos específicos, un analizador de embudos, que comprende el avance de los jugadores y algunas características más. Unity Analytics por tanto permite una buena visualización de los datos pero no hay nada mejor para el estudio de ellos que tener tu propio analizador y servidor adaptado a tus necesidades.

2.3. Videojuegos de Realidad Virtual

En la primera etapa de este proyecto, nos dedicamos a probar varios juegos que nos hicieran conocer algunos de los inconvenientes y de las ventajas que presenta actualmente el desarrollo de aplicaciones de RV. Los videojuegos son desde siempre el mejor campo de experimentación de nuevas ideas, y esto no es distinto en materia de RV. Este análisis del estado de la cuestión, nos sirvió para recoger ideas acerca de los mecanismos que podíamos llegar a utilizar en nuestros juegos de ejemplo, cómo mover al avatar, cómo interactuar con objetos, etc.

A continuación revisamos de manera esquemática algunos de los referentes que resultaron más útiles para nuestro trabajo, ya sea por cuestiones técnicas de RV o simplemente por servirnos para tomar ideas de ambientación o temática. Se resumen los objetivos, los mecanismos de juego y las conclusiones que obtuvimos tras analizar la experiencia que ofrece cada juego. Los videojuegos en cuestión son *HellVibe* (Eligogames, 2016), *Weigthless* (Schubert, 2014), *Q.U.B.E* (Toxic Games, 2016), *Pharaoh Puzzle* (GameTop, 2011) y *Aboard The Looking Glass* (Hoffman, 2015)

2.3.1. HellVibe

1. Objetivos

- Resolver el puzle.

2. Mecanismos de juego

- Girar el cubo
- Pasar la mano por encima de las piezas
- Empujar/tirar piezas internas del cubo mediante un toque
- Coger/Soltar piezas externas del cubo haciendo un gesto parecido a un pellizco

3. Conclusiones

- Mucha dificultad en el puzle, ya que no hay un tutorial inicial para entender el juego o pistas para ayudarte a avanzar. Hemos pensado en introducir un tutorial en nuestros juegos de ejemplo con el objetivo de ayudar al usuario a entender o finalizar el juego
- Interesante la mecánica de pasar la mano por encima de una pieza y que esta sobresalga

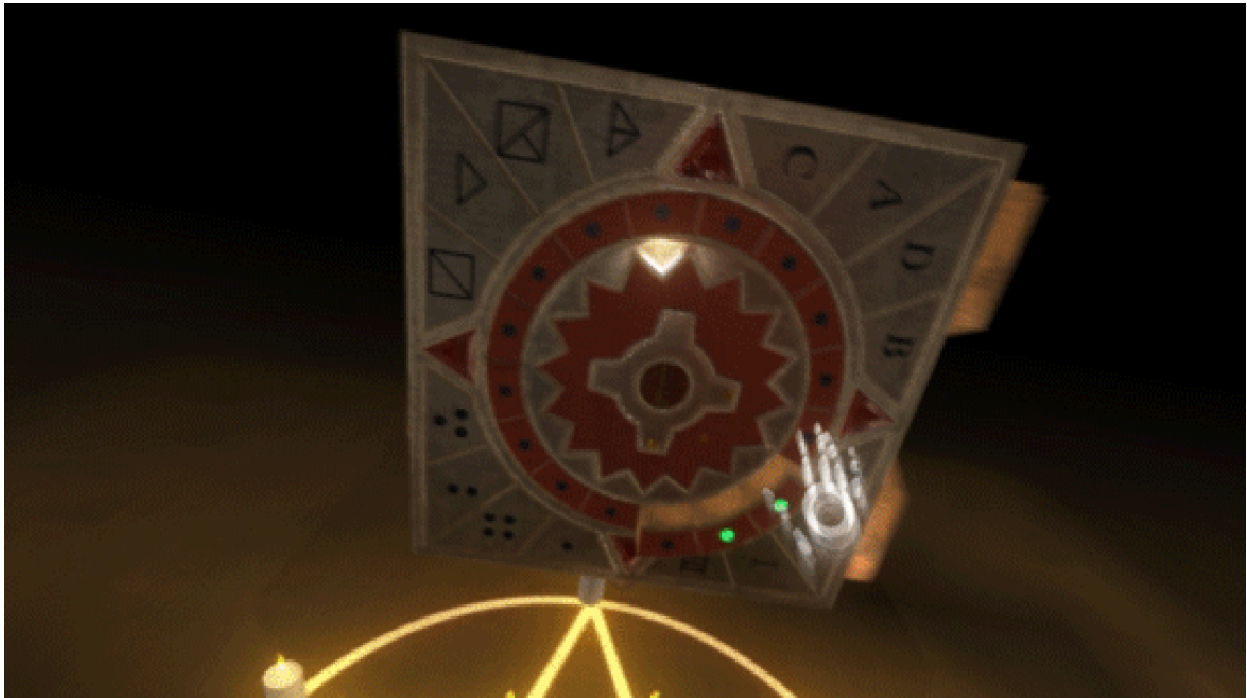


Figura 2.6: Una de las caras del cubo de HellVibe.

2.3.2. Weightless Remastered

1. Objetivos

- Explorar la nave
- Completar el modo historia

2. Mecanismos de juego

- Pulsar botones dándoles un toque
- Coger objetos haciendo el gesto de pellizcar
- Desplazarnos mediante propulsores en los guantes
 - Con la palma hacia delante (ambas manos) avanzas
 - Con los dedos hacia abajo y palmas hacia dentro retrocedes (sin embargo no sabemos por qué no terminó funcionando bien)

3. Conclusiones

- El modo historia no nos proporciona mucha información acerca de cómo avanzar en ella, solamente a través de audio. Esto puede perjudicar si el usuario está disperso o entretenido con el mundo de la RV, no tiene auriculares puestos o no entiende el idioma.
- No hemos visto acceso a ninguna ayuda en general, por lo que podemos incluirlo en nuestro juego

2.3.3. Q.U.B.E

1. Objetivos

- Recorrer los diferentes niveles interactuando con los cubos, cada uno tiene una función diferente, para completarlos. En las primeras fases, la meta del jugador es utilizar una combinación de los bloques para llegar a un punto de salida permitiendo de esta manera pasar a la siguiente habitación. A medida que el jugador progresa, las salas se volverán más difíciles y tendrán nuevos elementos con los que deberá interactuar

2. Mecanismos de juego

- Los bloques se distinguen por colores o por su función



Figura 2.7: Parte del juego *Weightless Remastered*.

- *Bloques rojos*: los cuales pueden ser extendidos o retraídos
 - *Bloques azules*: los cuales actúan como un trampolín cuando los tocas o te sitúas sobre ellos
 - *Bloques amarillos*: siempre en grupos de tres. Dependiendo de a cual apuntes se formará una escalera con tres peldaños de diferente forma
 - *Bloques verdes*: son los cubos que podremos desplazar o empujar
 - *Bloques violetas*: estos dan la posibilidad de rotar zonas del escenario
 - *Bloques transparentes* su función es impregnar de color al objeto que pasa a través de ellos teniendo que realizar diferentes mezclas de color
- Si los cubos tienen dos funciones como es el caso de los bloques rojos, éstos harán una u otra acción dependiendo de con que mano interactúes con ellos, es decir, en este caso, con la mano derecha se retraen y con la izquierda se expanden. Cada mano está asociada a un botón del ratón siendo la mano izquierda botón izquierdo y la mano derecha el botón derecho

3. Conclusiones

- El hecho de que sea un juego modular o basado en niveles nos permitía alejarnos del riesgo que pudiera llevar el crear una historia y no poder terminarla. De esta manera crearemos tantos niveles como nos dé tiempo y en caso de que alguno quedase a medias, no incluirlo sin afectar lo más mínimo al flujo del juego
- La posibilidad que nos da tener un dispositivo como el Leap Motion y que las manos sean la herramienta principal de acción en el juego, las cuales en QUBE tienen cada una funciones distintas. Esto nos hizo pensar en darle también distintas funcionalidades en el nuestro

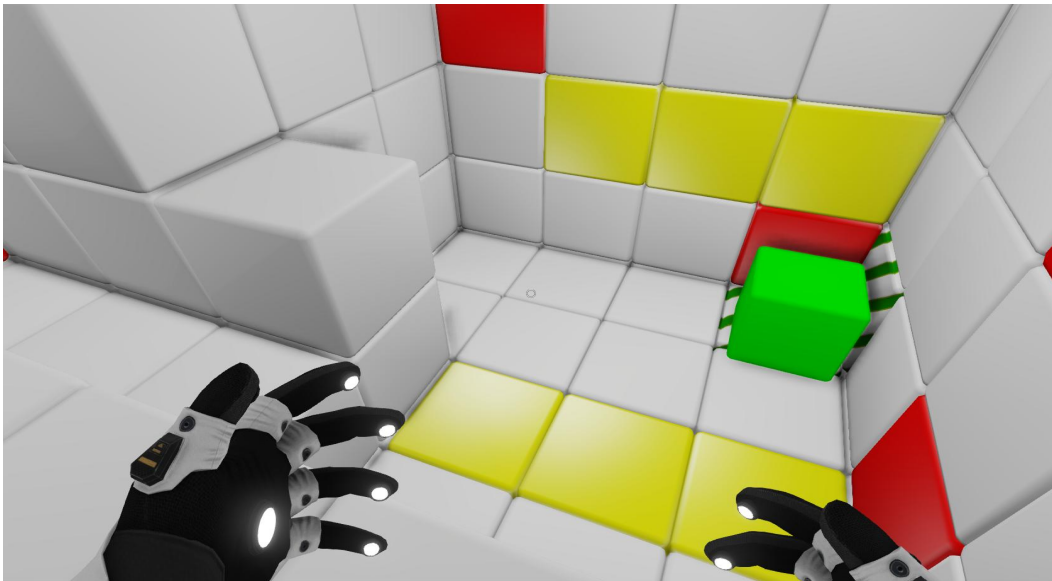


Figura 2.8: Imagen con cubos de distintos colores del juego QUBE.

2.3.4. Pharaoh Puzzle

1. Objetivos

- Encadenar al menos tres bloques del mismo color para recibir puntos
- Conseguir el máximo número de puntos en un tiempo limitado

- Algunos niveles de estos juegos suelen tener otro objetivo, como podría ser, por ejemplo, que ciertos objetos caigan a la base destrozando los cubos que haya por debajo

2. Mecanismos de juego

- Intercambiar posición de dos cubos, ya sea arrastrando uno hacia arriba, abajo, derecha o izquierda o bien seleccionando ambos (menos común)

3. Conclusiones

- De nuevo se presenta la idea de un juego modular y en este caso, de módulos más sencillos
- El mecanismo por el que los cubos se destruyen cuando ocurre cierta condición (al menos tres en raya del mismo color) nos llevó a darle una vuelta a nuestra idea de puzzle, haciéndonos llegar a la conclusión de que nuestros cubos también se debían destruir una vez usados. Sin embargo, pensamos en hacerlo de una manera distinta: que en vez de ser del mismo color, sean del color contrario. Esto se explicará más adelante, cuando contemos detalles de nuestros juegos.



Figura 2.9: Imagen de un nivel básico de Pharaoh Puzzle.

2.3.5. Aboard The Looking Glass

1. Objetivos

- Llegar al final del nivel resolviendo distintos puzles que se presentan, en los cuales tendremos que utilizar una mano u otra para hacerlo

2. Mecanismos de juego

- Dependiendo de la mano que tengamos entre el objeto y nosotros, veremos una cosa u otra. La mano izquierda dejará ver el pasado, mientras que la mano derecha nos mostrará el futuro

3. Conclusiones

- De nuevo aparece la idea de un darle una función distinta a cada mano. En este caso, se juega con el tiempo y lo que se verá a través de cada una. Nosotros nos decantamos más por la segunda opción y pensamos situaciones en las que pudiéramos aprovechar esta idea



Figura 2.10: Mano izquierda descubriendo una parte oculta.

2.4. Retos actuales de la Realidad Virtual

En esta sección haremos mención a los problemas e inconvenientes para la experiencia del usuario que tiene actualmente la RV. Primero explicaremos aquellas que tienen que ver simplemente con los dispositivos en general y con las herramienta de desarrollo, las propias limitaciones tecnológicas del medio, y a continuación los inconvenientes encontrados al probar algunos de los juegos comentados en la sección 2.3. Haremos referencia al mismo tiempo a posibles soluciones que se pueden llegar a implementar para evitar o mitigar de alguna manera los problemas de estas aplicaciones aún tan experimentales.

2.4.1. Retos a nivel tecnológico

Entre los principales problemas que surgen al usar los visores de RV y que consiguen que algunos usuarios acaben quitándose el dispositivo de encima son los mareos, los sudores fríos y las nauseas que aparecen al poco tiempo de su uso, el llamado *Simulator Sickness*⁸. Esto se debe a que nuestro cerebro esté acostumbrado a que los sentidos actúen con una perfecta armonía y sincronización, y al ponernos el casco de RV, si el juego no está cuidadosamente diseñado se pueden llegar a generar incongruencias en esa percepción, como por ejemplo con los cambios bruscos y artificiales del movimiento del avatar en el juego.

Sin embargo ya hay empresas que están investigando en cómo reducir estos síntomas de distintas maneras al igual que se proponen “trucos” para evitarlos a la hora de introducirte en una aplicación de RV. Por ejemplo, la revista Wired se hace eco del trabajo de unos investigadores de la Universidad de Purdue que creen haber encontrado una forma de reducir los efectos físicos negativos anteriormente descritos haciendo visible algo que todos tenemos en frente de la cara y que nos sirve de referencia casi inconsciente: la nariz. De hecho, el profesor asistente en el Departamento de Tecnología de Computación de dicha universidad, David Whittinghill, asegura que han llegado a reducir la aparición estos efectos en un 13.5 %.

⁸En la propia página web de desarrolladores para Oculus hay una sección referente a este tema. www.developer.oculus.com/design/latest/concepts/bp_app_simulator_sickness/

Por otro lado tenemos a los usuarios con problemas de visión, los cuales no se sienten cómodos con los visores de RV, si este les impide usar sus gafas o tener que llevar puestas las gafas más el visor, en una configuración un tanto engorrosa. El abuso de estos dispositivos puede llegar a generar vista cansada así como agravar los problemas ya existentes de visión que tengamos.⁹. Otro factor a tener en cuenta es la resolución del dispositivo que se esté usando. En nuestro caso hemos estado usando dos versiones diferentes de las Oculus Rift. La segunda versión para desarrolladores (DK2), que tiene una resolución de 1920 x 1080 píxeles (960 x 1.080 píxeles por ojo) a 75 Herzios con un campo de visión de 100 grados y la versión para el consumidor (CV1) con una resolución de 2.160 x 1.200 píxeles (1.080 x 1.200 píxeles por pantalla) a 90 Herzios.

2.4.2. Retos a nivel de aplicación

Los problemas que hemos encontrado al probar los distintos juegos de RV comentados en la sección 2.3 surgen de la poca experiencia que hay en el diseño de videojuegos para este nuevo medio, del poco interés que existe todavía por arriesgar en el desarrollo de un título grande (AAA) y del interés, a veces oportunista, de querer crear un producto lo más simple y divertido posible para tomar posición en este mercado emergente pero sin explotar del todo la tecnología.

Un caso bastante claro es el de HellVibe (mencionado previamente en la sección 2.3.1) en el que nada más empezar te encuentras en una sala con un cubo en el que cada una de sus caras tiene un puzzle diferente, lo que te hace sentir que hay demasiadas variables y piezas en cada cara. En ningún momento nos dio la sensación de que los movimientos que estábamos haciendo eran los correctos o que estábamos avanzando en el puzzle y tampoco había ninguna explicación o pista a la que acudir por si necesitabas ayuda, opción que un juego de puzzles debería intentar tener para hacer la experiencia más amena y grata, sin necesidad de cambiar mucho la dificultad del nivel.

⁹Uno de los artículos en los que se hace referencia a este problema es el siguiente: https://essilorusa.com/content/essilor-usa/en/newsroom/news/virtual_reality_bad.html

Un caso contrario al de HellVibe, en el que sí recibes algo de ayuda y conoces el objetivo del juego, es el de Aboard The Looking Glass (véase en la sección 2.3.5). Durante la breve duración del juego, se sabe a grandes rasgos lo que hay que hacer para avanzar. En primer lugar, el juego tiene una historia y una trama, por lo que la inmersión al juego quizá sea de las mayores que hemos visto. En segundo lugar, nada más empezar ya te dan información acerca de lo que tienes que hacer y de lo que está pasando, como se puede ver en la Figura 2.10 en la cual se contempla el mensaje “Point to activate” (información acerca de lo que tienes que hacer) y encima de éste hay otro mensaje en el que pone “Unlocking” (información acerca de lo que está ocurriendo).

De esta manera, la sensación que vas obteniendo durante el juego es que éste se hace más entretenido sin necesidad de disminuir la dificultad del mismo. Otro punto a favor es que al mantenerte alerta deja menos tiempo al usuario para pensar en las incomodidades asociadas a la RV, como el peso y el calor que genera el visor, el ajuste de las correas, o la siempre temida posibilidad de golpearse con los objetos que hay alrededor del jugador en el mundo real, mientras intentas moverte en el virtual.

Capítulo 3

Objetivos y especificación

El objetivo principal de este proyecto es conseguir desarrollar un sistema que permita analizar las acciones que realiza el usuario durante una experiencia de Realidad Virtual avanzada, con seguimiento de movimiento del visor, de las manos y reconocimiento gestual de las manos y los dedos. A través de la recogida de datos de los distintos sensores de Oculus Rift y Leap Motion, se crearán una serie de tablas y gráficas que podrán ser mostrados después a los investigadores de RV que estén trabajando en hacer experimentos con usuarios reales en las aplicaciones que ellos elijan. Estas estadísticas se mostraran en una sitio web en forma de contenido descargable para que los investigadores pueda hacer el tratamiento que crean conveniente, realizar sus estudios y sacar sus conclusiones sobre cómo los usuarios se desenvuelven en la experiencia o aplicación que esté tratando de analizar. El sitio web, al que denominaremos *Haptic Analytics*, también hará un procesamiento básico de información para calcular algunas variables como la media, los porcentajes, o establecer *rankings* de usuarios según el tiempo en realizar una tarea, por ejemplo. Para representar los datos necesitaremos una parte frontal (o *front-end*) del sitio web mientras que para procesarlos y almacenarlos debidamente habrá que desarrollar una parte trasera (o *back-end*).

Para ilustrar y probar debidamente el funcionamiento de este sistema, es imprescindible crear alguna aplicación donde poder probarlo. En este proyecto vamos a desarrollar dos aplicaciones diferentes, dos juegos de RV que funcionarán en primera persona para dar al usuario una sensación mayor de “presencia”, y donde se usarán las manos para resolver enigmas y puzles. La idea de estos juegos es que sean modulares para poder añadir más o menos niveles según el tiempo disponible, siguiendo una planificación propia de una metodología ágil. Otro motivo importante por el que haremos juegos modulares es para evitar dejar a medias una narrativa o una experiencia más compleja que no llegásemos a terminar, al fin y al cabo el objetivo de estos juegos no es otro que demostrar que *Haptic Analytics* funciona correctamente.

3.1. Objetivos

Los objetivos generales del proyecto a la hora de desarrollar el sistema *Haptic Analytics* son los siguientes:

- Facilitar la recogida de datos de la interacción del usuario con la aplicación de RV, desarrollando un complemento software que una vez añadido a la aplicación simplifique y automatice en la medida de lo posible el proceso
- Establecer una interfaz de programación de aplicaciones (API) y protocolo de envío de los datos recogidos a un servidor a través de la World Wide Web
- Ordenar los datos del servidor y calcular variables estadísticas que reflejen los aspectos del comportamiento de los usuarios que más interesen a los investigadores en RV
- Permitir la gestión y visualización de toda esta información mediante un sitio web con acceso controlado y específicamente diseñado para estos investigadores

- Desarrollar dos juegos de RV diferentes que sirvan como aplicaciones de referencia para el uso y despliegue del sistema propuesto. Para acotar el ámbito tecnológico de estas aplicaciones nos centraremos en la combinación de Oculus Rift con Leap Motion.

3.2. Especificación de requisitos software

En las próximas subsecciones se especifican los requisitos software necesarios para la herramienta *Haptic Analytics*, así como para los juegos de ejemplo *Cube Board* y *Escape Room*.

3.2.1. Obtención de datos

Desde un punto de vista temporal, vamos a considerar el tiempo inicial (T_i) cuando arranca la aplicación y el tiempo final (T_f) el momento en el que esta termine (por ejemplo, cuando el jugador resuelve un puzzle). Podemos estudiar la duración de cada actividad del usuario en tiempos absolutos o relativos, normalizando las duraciones de todos los usuarios utilizando un porcentaje.

Estos son algunos tipos de datos que tiene interés obtener para el análisis de una aplicación de RV:

- A través del objeto *Hand* que ofrece la API de Leap Motion, por ejemplo, es posible:
 - Saber qué mano toca un objeto o se ve, si es la izquierda o la derecha
 - Obtener información posicional de cualquiera de los dedos de cada mano
 - La posición de cada una de las manos junto con la orientación de la palma, la muñeca y los dedos
 - El vector de dirección hacia donde apuntan los dedos y la palma

- El ángulo de agarre, yendo desde 0 hasta 180 grados. Este ángulo es el ángulo entre la dirección de todos los dedos excepto el pulgar y la dirección de la mano. El ángulo para la mano abierta será de 0 grados mientras que los 180 grados serán cuando esté cerrada
 - La fuerza de agarre, yendo desde 0 hasta 1, siendo el valor 0 la mano abierta y 1 la mano cerrada
 - La velocidad con la que se mueve la mano en milímetros por segundo. Con ella podremos sacar la derivada y obtener la aceleración, aunque puede ser sensible al error en caso de querer hacer cálculos de precisión
 - El tiempo que han estado visibles las manos
- También hay atributos útiles e interesantes del brazo que podemos obtener, como:
- La posición del codo
 - La posición de la muñeca
 - El vector de dirección del codo hasta la muñeca
 - Otros en principio no tan interesantes como puede ser la posición del punto medio entre codo y muñeca, tamaño del brazo, el extremo del hueso más cercano a la punta del dedo, etc.

3.2.2. Medidas y variables estadísticas

Además de la información “cruda” que proporcionan los cascos de RV y los dispositivos de reconocimiento de gestos, un investigador de RV estará interesado en otras medidas y en algunas variables estadísticas básicas que puedan describir cómo se relaciona un usuario con la aplicación. Algunas de estas medidas y variables estadísticas que se pueden enviar son:

- Número de veces que el usuario mira a ciertos puntos del nivel

- Separar los objetos por etiquetas y de esta manera calcular el tiempo que se pasa el usuario prestándoles atención. Los objetos se podrían separar entre estáticos y dinámicos
- Número de veces que el usuario toca cada uno de los objetos
- Tiempo que un jugador tarda en pasarse los tutoriales
- Tiempo que un jugador tarda en terminar el nivel
- Forma de terminar dicho nivel, es decir, con éxito o sin él
- Tiempo que pasa el jugador con las manos activas/visibles (nos lo da una variable del objeto *Hand* proveniente de la API de Leap Motion)
- Normalización de los tiempos según haya tardado en terminar los niveles, enviado como un porcentaje, por ejemplo

3.2.3. Dinámica del puzle *Cube Board*

Este puzle consiste en un único escenario, una sala, con un tablero vertical en medio, bastante cercano y con un sistema giratorio, que permite girarlo y acceder a las dos caras del tablero, estando cada una de ellas divididas en casillas.

En una de las caras hay piezas de colores dibujadas, y en la otra piezas similares pero móviles, con forma de cubos. Las piezas corresponden a elementos, que serán fuego, agua, vegetación y vacío (sin elemento). Generalmente operamos sobre la cara del tablero que tiene las piezas móviles, aunque se puede girar el tablero para ver cómo están situadas las piezas por la otra cara. El objetivo es situar las piezas móviles opuestas (agua contra fuego, etc.) justo en la posición opuesta respecto al eje horizontal del tablero de otra pieza igual pero que esté en la otra cara. Cuando se produzca esta relación entre un par de piezas, las piezas estáticas dibujadas irán “desapareciendo”.

Por ejemplo, suponiendo que tenemos un tablero de 6 x 6, si en la cara estática una pieza de fuego está en la posición (2, 6), siendo el primer número el número de fila (eje vertical) y el segundo el número de columna (eje horizontal), en la otra cara del tablero, la cara dinámica, deberíamos colocar una pieza móvil *de agua* justo en *la posición (2, 1)* para que desaparezca el dibujo de la cara estática. Una vez hayamos conseguido hacer desaparecer todos los dibujos, se habrá pasado el nivel.

Todo ello hay que hacerlo teniendo en cuenta que cada pieza que hacemos desaparecer se va acumulando en la torre correspondiente de un conjunto de tres torres que tenemos a nuestro lado. Si alguna de esas torres se llena demasiado más que las otras, perdemos, con lo que mientras estamos resolviendo el puzle hay que intentar ir alternando de “color” de la jugada a cada momento, para no desequilibrar las torres.

3.2.4. Elementos del escenario

A continuación explicaremos cómo están compuestos los niveles, como por ejemplo el escenario del tablero del primero juego que es el objeto principal sobre el que actuaremos junto a las piezas.

Esta sala contiene un tablero de doble cara, un mecanismo que da la vuelta el tablero y otro con el que empezar de nuevo el nivel.

La primera cara del tablero que ves nada más empezar el nivel, la denominaremos cara A o cara dinámica, mientras que la cara contraria será la cara B o cara estática. La posición de las piezas en ambas caras se generará de manera aleatoria siendo la diferencia entre ellas que en la estática no se podrán mover las piezas como su propio nombre indica, si no que servirá de referencia para actuar en la otra cara, en la dinámica.

Algunos cubos de la cara dinámica tendrán el símbolo oculto, es decir, que no se verán a simple vista si no que reaccionarán ante el movimiento del brazo izquierdo para poder visualizar el símbolo del cubo. Hay dos botones que están a los lados del tablero, uno rojo y otro verde. El rojo, que estará en la parte derecha, servirá para dar la vuelta al tablero, mientras que el verde, que está en la izquierda, servirá para reiniciar el nivel.

3.2.5. Formas de interacción del jugador

Las acciones que podremos realizar a lo largo del primer juego son las siguientes.

- Mover la cabeza para observar la sala y sus elementos
- Apuntar hacia el objeto con la mano izquierda, la cual revelará los símbolos ocultos
- Pasar la mano derecha por encima de las piezas cúbicas hará que sobresalgan un poco para saber que estás apuntando o seleccionando esa pieza
- Coger una pieza. Cerrar la mano como si estuviéramos agarrando un objeto mientras apuntamos a un cubo servirá para cogerlo
- Soltar una pieza. Con un cubo cogido, si se abre la mano lo soltaremos. En caso de soltarlo sobre la misma posición simplemente se colocará como estaba antes de cogerlo. En cambio, si lo soltamos encima de otro cubo pueden pasar dos cosas. En caso de ser un cubo con un símbolo y soltarlo en una posición en la que en la cara opuesta haya un símbolo contrario, el cubo que había y el que teníamos en la mano desaparecerán. En caso contrario simplemente el cubo que tenemos cogido se colocará en la posición a la que estábamos apuntando y automáticamente se agarrará el cubo que había en esa posición
- Girar el tablero. Pulsando el botón rojo que está a la derecha del tablero, este dará una vuelta de 180 grados sobre el eje vertical
- Reiniciar tablero pulsando el botón verde que está a la izquierda de este

3.2.6. Tutoriales, menús e interfaces

A continuación definiremos los tutoriales de que dispondrá nuestro juego para ayudar al usuario a conocer el funcionamiento del mismo, junto a los menús que facilitan la accesibilidad.

- Tutoriales

Como mencionamos anteriormente en el Capítulo 2, nos dimos cuenta probando otros ejemplos, demos y juegos que la falta de tutoriales que explicasen un poco las acciones posibles u objetivos del juego hacían la experiencia en RV bastante frustrante. Por ello decidimos crear una serie de tutoriales que ayudasen e hicieran conocer al usuario las acciones y movimientos que se pudiesen hacer durante el juego.

La primera parte del tutorial consistiría en explicar los elementos de la sala además de enseñar al usuario como agarrar y soltar los cubos. De esta manera también aprendería a destruir las piezas de símbolos opuestos para poder pasarse el nivel. La segunda parte consistiría en enseñar al jugador que dispone de una herramienta para descubrir símbolos ocultos. De esta manera tendría que encontrar piezas con el símbolo oculto y conseguir hacerlas desaparecer para pasarse el tutorial. Una vez hubiera completado el tutorial pasaría a jugar el juego propiamente dicho.

- Menú principal En el caso de que la implementación de los tutoriales y el juego fuera afín a nuestros gustos y tuviese una jugabilidad correcta crearíamos una serie de menús para poder acceder a los distintos niveles desarrollados.

El menú que nos encontraremos nada más iniciar el juego o al salir de un nivel tendrá los siguientes botones:

- Comenzar: el cual te introducirá en primer lugar al tutorial en caso de ser la primera vez que se juegue y al último nivel completado en caso contrario

- Seleccionar nivel: te dejará escoger entre ir a los tutoriales o a cualquiera de los niveles
 - Estadísticas: te llevará a la página web donde se mostrarán las estadísticas del usuario en el juego
- Pausa

Este será el menú que saldrá al pulsar el botón concreto de pausa mientras estamos en el juego o en el tutorial.

- Volver al juego: similar a cancelar la pausa o continuar
 - Volver al menú principal: te devuelve a la primera pantalla
 - Repetir nivel: empiezas de nuevo el nivel. Generará otro tablero con disposición de piezas aleatoria
 - Seleccionar nivel: la misma función que Seleccionar nivel del menú principal
- Interfaces

Con estas interfaces nos referimos a los cuadros de texto que habrá en los tutoriales y distintos niveles explicando el objetivo de estos. Los mensajes se mostrarán en la parte superior de la pantalla y estarán colocadas en el objeto de la cámara, es decir, en caso de que el jugador desvíe su atención a cualquier otra parte de la pantalla que no sea el tablero, el mensaje seguirá apareciendo. El mensaje no se mostrará entero al instante si no que se mostrará palabra a palabra

3.2.7. Sistema de secuencias

- Desde el menú principal:

Se podrá acceder a las distintas partes comentadas en la subsección anterior. Si se pulsa sobre el botón “Comenzar”, el juego empezará desde los tutorial y una vez se completa el último se pasará al primer nivel.

Si seleccionamos la opción de "Seleccionar nivel", se dará la posibilidad de ir tanto a los tutoriales como a cualquier nivel, por ahora sólo el nivel principal. Una vez se complete, también se mostrarán los resultados conseguidos y se pasará al nivel siguiente. Si el nivel resuelto es el último, se volverá al menú principal.

- Desde el menú de pausa:

Para poder acceder a este menú deberemos estar jugando el tutorial o el nivel. Cumpliendo esta condición, los flujos de secuencias que pueden existir son los siguientes:

Pulsando el botón "Volver al juego" se cerrará el menú de pausa y podremos continuar jugando desde donde lo pausamos.

En caso de presionar "Volver al menú principal", nos dirigiremos al inicio del juego, es decir, a la primera pantalla que se ve nada más ejecutarlo, pudiendo acceder a todas las otras opciones explicadas anteriormente.

Si eligiésemos la opción "Repetir nivel" volveremos a empezar, pero esta vez el tablero se generará de nuevo y será distinto al anterior. Se vuelve a iniciar la estancia del nivel.

Por último, si escogemos "Seleccionar nivel" podremos acceder al tutorial o acceder a otro nivel.

3.2.8. Dinámica del videojuego *Escape Room*

El nivel dispone de varias salas en el que el jugador deberá descubrir como salir de ellas, mediante el uso de las manos y la lógica. La primera sala será una celda medieval en la que para escapar tendrá que tirar una piedra a una antorcha que esté fuera de la celda, de este modo la sala se verá iluminada desde otro ángulo y el jugador podrá apreciar que hay baldosas que sobresalen un poco de la pared y las cuales tendrá que pulsar el jugador para poder salir de la celda.

En la segunda sala, nos encontramos con un piano que posee cinco teclas enumeradas del uno al cinco. Para salir de esta habitación, el jugador tendrá que seguir una serie de pistas para descubrir la combinación que hay que ejecutar tocando de teclas del piano y así abrir una puerta secreta que lleva a la salida.

3.2.9. Envío, almacén y visualización de datos

Tras el envío de datos, una vez llegue al servidor, la información será o no tratada, dependiendo de las necesidades del sistema y posteriormente será almacenada.

Una vez estén almacenados los datos el usuario podrá visitar el sitio web de *Haptic Analytics* en la que se mostrarán todos los datos obtenidos a lo largo de las sesiones de juego.

El servidor será montado principalmente en uno de los ordenadores de los componentes del grupo con la tecnología escogida y una base de datos que funcionarán como *back-end*. En cuanto a la parte de *front-end* se intentará hacer con una tecnología bastante moderna. De esta manera tendríamos separadas claramente cada una de las capas de la aplicación y se utilizarían tecnologías no necesariamente muy usadas en el ámbito universitario.

Capítulo 4

Metodología y herramientas

Para la realización del proyecto seguimos principalmente una estructura de descomposición de trabajo (EDT) que se adaptaba bastante a la forma en la que después fuimos separando el proyecto y en la que íbamos haciendo pequeñas entregas o muestras al director y al codirector del trabajo. Tomamos algunas ideas para el proceso de desarrollo de las metodologías ágiles como por ejemplo la toma de decisiones a corto plazo a través de reuniones periódicas, y la planificación periódica tanto con el director y el codirector como entre nosotros. Otro de los aspectos clave en el que nuestro desarrollo estuvo muy influenciado por esta filosofía ágil es en la aceptación de nuevos requisitos que frecuentemente aparecían, según íbamos comprendiendo mejor en qué consistían los objetivos del proyecto, teniendo nuevas ideas o descubriendo mejores formas de avanzar en el desarrollo.

4.1. Metodología

La estructura anteriormente mencionada la separamos en tres secciones. La primera sección serían los juegos que creamos para probar nuestra herramienta, en segundo lugar estaría el sitio web como el *front-end* y el *back-end* de la aplicación. Cada uno de los integrantes nos dedicábamos especialmente a una parte en concreto, pero obviamente para conectar las tres partes y hacerlas funcionales todos tuvimos que utilizar y desarrollar partes de los demás. Sabiendo entonces como estábamos estructurados intentamos dirigir las reuniones de manera que apareciesen nuevas tareas en cada una de las partes y así poder ir avanzando en paralelo los tres componentes del grupo.

Estas reuniones en las que principalmente aparecían los nuevos hitos se hicieron aproximadamente cada dos semanas y siendo estas presenciales junto con los coordinadores del proyecto. El propósito principal era que para la nueva reunión estuviesen realizados los objetivos planteados de la anterior. Sin embargo, entre una y otra sesión, los componentes del grupo realizábamos encuentros vía Skype¹ para definir algún detalle, como por ejemplo nombres de variables que se pasan en los mensajes entre cliente y servidor, o si necesitábamos una nueva función para recoger o enviar datos. Además Skype tiene una función que permite compartir pantalla y podíamos ver en directo la parte de los demás sin necesidad de descargar el proyecto y ejecutarlo en Unity, en el caso de los juegos, o de hacer unificar las ramas del repositorio que contiene el *front-end* y el *back-end* ya que podía haber tareas a medio hacer.

Principalmente uníamos el proyecto alojado en los repositorios cada vez que añadíamos una nueva funcionalidad y el día antes a la reunión presencial para así poder mostrar la última versión.

¹Skype: <https://www.skype.com/es/>

4.2. Plan de trabajo

Debido a que el visor de realidad virtual se compartían con otros grupos, el director del proyecto creó un calendario para organizarnos con el material y que de esta forma todos pudiéramos hacer uso de ellas. Esto también nos sirvió para organizarnos nosotros como grupo e ir dedicándonos más a la memoria o al desarrollo de la herramienta y de los juegos. Durante el primer trimestre, solo había un equipo de Oculus Rift por lo que íbamos rellorando la memoria todo lo posible hasta tener el caso de RV disponible y ser capaces de probar el juego con ellas. Una vez las teníamos nos juntábamos para seguir con el desarrollo del juego. Sin embargo, el departamento adquirió otro equipo de RV y a partir de ahí cada uno de los grupos teníamos una de las gafas, por lo que internamente, nos acabamos organizando de manera distinta.

Como el desarrollo en RV no está muy avanzado nos era más complicado obtener buenos ejemplos en los que poder influenciarse o inspirarse. También, dado que las acciones que se pueden realizar en los juegos de RV junto al Leap Motion son algo limitadas y la existencia de las consecuencias que tiene la RV (sección 2.4.1), intentamos simplificar las nuestras lo máximo posible. De esta forma conseguiríamos que no nos llevase más trabajo de la cuenta crear el juego. Respecto al desarrollo del juego, hicimos la mayor parte durante la segunda mitad del año lectivo, una vez fijamos los requisitos, parámetros, objetivos y herramientas a conseguir o usar.

Para la parte de la creación del juego pensamos que el hecho de que fuera modular y sin una historia concreta era la mejor manera de no comprometernos a que quedasen cabos sueltos y que no diese la sensación de no estar terminado. Además, de esta manera podíamos extender el juego en el futuro añadiendo nuevos niveles. Durante el segundo cuatrimestre se hizo por lo tanto la herramienta principal del proyecto, el juego donde utilizarla, y la implementación del *front-end* y *back-end*.

El plan de trabajo que seguimos durante el desarrollo del proyecto se hizo en las primeras reuniones a principios de curso junto a los coordinadores, en las que revisamos cómo estaría organizada la memoria a grandes rasgos y en qué partes. De esta manera pudimos dividir el trabajo por trimestres dando lugar a las fechas de entrega y objetivos que tendríamos a lo largo del curso. En un principio se acordó entregar los tres primeros puntos de la memoria a la vuelta de las vacaciones de Navidad. Sin embargo, la semana previa a este mismo periodo tendríamos que tener realizado algo de material, ya fuera código o memoria para poder exponer lo que llevábamos hasta la fecha a los demás grupos de alumnos trabajando con el mismo director. Esto se hizo con la intención de saber cómo íbamos de avanzados respecto a los otros grupos de alumnos y para que los tutores pudieran comprobar si nos estábamos quedando atrás o si no lo estábamos enfocando correctamente.

Para el segundo cuatrimestre se acordó la entrega de una versión poco desarrollada pero funcional del código, el cual debía incluir tanto la herramienta como el juego en el que probarla. Las partes de la memoria a las que se asocia son los Capítulos 4 y 5 y la fecha de entrega para este hito fue entre la última semana de marzo y la primera de abril, justo antes de las vacaciones de Semana Santa.

Por último se dejaron el resto de capítulos para entregar los primeros días de junio, junto con una revisión general de las secciones anteriores. De esta manera tendríamos tiempo entre la entrega a los directores y la entrega oficial de la memoria y el proyecto para revisar todos los posibles detalles que no concordasen y corregir errores de forma, además de tener la posibilidad de hacer distintas pruebas de la presentación del Trabajo de Fin de Grado.

4.3. Herramientas

Las herramientas que utilizamos para la comunicación entre los componentes y coordinadores del proyecto, el alojamiento de las diferentes secciones y además de la realización de la memoria fueron las siguientes:

4.3.1. Creación artística

Debido a que no encontrábamos algunos objetos, texturas o materiales que nos gustasen para el acabado de nuestro juego, decidimos crearlos nosotros con una herramienta generadora de modelos en tres dimensiones como es el caso de Blender, en algunos casos como no nos dio tiempo tuvimos que hacer uso de modelos prediseñados y adaptarlos a nuestros juego.

4.3.1.1. Blender

Blender (Blender Foundation, 1995) es un programa dedicado al modelado, iluminación, renderizado, animación, simulación y creación de gráficos tridimensionales. Lleva siendo de código abierto desde 2002 y entre sus muchas utilidades, sirve para la edición no lineal de vídeo y hasta puede usarse como motor de videojuegos.

Concretamente hemos usado esta herramienta para crear la textura del tablero, y generar los cubos que lo componen en el primer juego desarrollado, *Cube Board* (véase Figura 4.1). También lo utilizamos para crear el escenario del videojuego *Escape Room*, aunque hemos optado por conseguir la gran mayoría de los modelos 3D de otras fuentes.

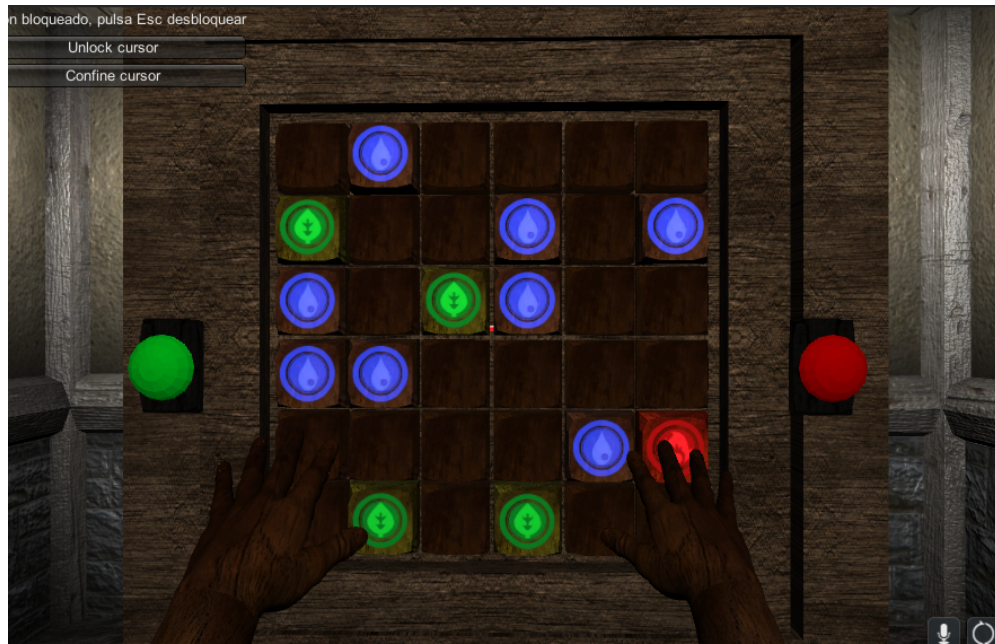


Figura 4.1: Imagen del primer juego creado con modelos desarrollados por nosotros

4.3.1.2. Adobe Photoshop

Editor de gráficos rasterizados usado principalmente para el retoque de fotografías y gráficos. Se ha usado principalmente para retocar imágenes.

4.3.2. Comunicación

4.3.2.1. Skype

Este software nos ha permitido realizar las reuniones entre los integrantes del grupo a través de Internet. Escogimos esta opción ya que todos la teníamos previamente instalada, es gratuita (al menos en el uso que nosotros le hemos dado) y nos permitía compartir las pantallas, librándonos de describir algunas dudas o problemas que teníamos y poder solucionarlas con un simple vistazo.

4.3.2.2. Slack

Esta herramienta de comunicación no es tan conocida pero es realmente útil. La usamos como recomendación del director del proyecto ya que él la utiliza para la comunicación con sus alumnos y con los distintos proyectos de los que se hace cargo. La ventaja que ofrece este sistema es que ofrece salas de chat organizadas por temas, dando la posibilidad de hacer tanto grupos privados como mensajes privados. Además tiene un buscador que permite acceder a todo el contenido de los grupos en los que estás inscrito o asociado. Otro punto fuerte de Slack es que integra una notable cantidad de servicios a terceros como pueden ser Google Drive, GitHub o Trello, entre otros.

4.3.3. Alojamiento compartido y control de versiones

Para el alojamiento del proyecto necesitábamos una buena plataforma de control de versiones, y un sistema general para compartir información.

4.3.3.1. GitHub

Es la herramienta que habíamos aprendido a usar de forma básica durante la carrera. Además nos permitía hacer ramas del proyecto principal para poder desarrollar en paralelo sin riesgo a perder el contenido en caso de ser borrado localmente. De esta forma se pueden ir unificando las ramas cada poco tiempo para tener todos los componentes el mismo código y trabajar a partir de un punto concreto.

4.3.3.2. TortoiseGit

Para facilitar la ejecución de los comandos de Git (*commit*, *push*, *pull*, *merge*, etc.) sobre el proyecto decidimos usar este cliente gráfico para Windows que lo hace todo mucho más sencillo y legible para el usuario.

4.3.3.3. Google Drive

Como sistema de alojamiento y compartición de ficheros utilizamos Google Drive, ya que para ficheros de documentación o para crear las actas de las reuniones que íbamos teniendo era mucho más cómodo. Además también fuimos alojando las imágenes que usaríamos en la memoria o en las distintas presentaciones que hicimos a lo largo del año para mostrar el avance del proyecto. Hemos elegido esta herramienta sobre otros servicios de alojamiento en la nube debido a que posee herramientas de edición de documentos online con las que estamos familiarizados los miembros del proyecto y porque poseemos almacenamiento ilimitado gracias a que contamos con cuentas de la Univesidad Complutense de Madrid, que tiene un acuerdo con Google para ofrecer ventajas a los alumnos de dicha universidad.

4.3.4. Realización de la memoria

Finalmente, aunque se tomasen apuntes en Google Docs era necesario contar con una buena herramienta para la redacción y la maquetación final de la memoria del trabajo.

4.3.4.1. LaTeX y Overleaf

Utilizamos esta herramienta de escritura y edición de documentos como aplicación principal para desarrollar esta memoria. Overleaf es una herramienta *on-line* del conocido sistema de composición de textos LaTeX y que además permite la implantación de plantillas como por ejemplo la que está siendo usada, que es la que propone la Facultad de Informática de la Universidad Complutense de Madrid.

Capítulo 5

Haptic Analytics: Análisis, diseño e implementación

En este capítulo abordaremos el análisis y diseño, tanto de la parte relacionada con la interacción del cliente y el sitio web, como de la parte correspondiente con la interacción del usuario y los juegos desarrollados.

En primer lugar, en el apartado 5.1 haremos un análisis de cómo se tiene previsto usar el sistema. A continuación se mostrarán los diagramas de caso de uso que reflejarán lo contado previamente.

El siguiente apartado, 5.2, contendrá todo aquello que tiene que ver con el diseño de la herramienta, mostrando las relaciones que tienen las entidades y tablas de la base de datos entre sí y explicando la arquitectura en general. De esta manera abarcaremos todo lo relacionado con la comunicación entre las partes señaladas al principio del capítulo.

Por último, en el apartado 5.3 indicaremos la estructura y las tecnologías usadas en cada capa, así como detalles de la implementación de la aplicación.

5.1. Análisis

Como se pensó desde un principio, una función principal de nuestra herramienta debe ser la recogida de datos para su posterior visualización y estudio. Esto exigía mínimo una conexión entre el generador de esos datos (los juegos implementados) y una base de datos que los almacenase junto a un servicio de *back-end* que hiciera todo el procesado y validación de los mismos. La meta era hacerlo lo más genérico posible para que añadiendo unos *scripts* a cualquier juego creado en Unity se pudiera conectar con nuestra herramienta, a ser posible simplemente con el hecho de rellenar ciertas variables ya sea de manera manual o dinámica en esos *scripts*. De esta manera nosotros los almacenaríamos en nuestra base de datos para su futura visualización y estudio.

Un caso de variable introducida manualmente sería por ejemplo el nombre que tendrá el juego y al cual se le asigna la sesión, mientras que un caso de variable añadida dinámicamente sería por ejemplo todos los objetos que han sido tocados, cuántas veces y que además quieras almacenar, es decir, aquello que el investigador crea conveniente enviar. En el apartado de Diseño, 5.2 se mostrará la estructura que sigue el JSON que se envía al *back-end*.

La otra función principal que tiene nuestra herramienta es la visualización y estudio de los datos, la cual no ha sido la más difícil de hacer pero sí la que más cambios y puntos de vista ha tenido a lo largo del año.

En esta herramienta se hace una división de usuarios entre uno corriente y otro que se denominará Investigador. La diferencia entre ellos es que el Investigador podrá crear y ver Experimentos y Juegos para estudiar los datos que están relacionados con ellas mientras que el usuario solo podrá ver información acerca de los Juegos a los que ha jugado.

Desde el apartado de Experimento, el Investigador podrá acceder a los detalles de una misma, donde saldrán tanto la información básica (nombre e identificador único, el cual se entregará a los Sujetos más adelante) como una tabla con las sesiones jugadas y una gráfica que mostrará los juegos jugados y el número de veces que han sido jugados. En cada fila de la tabla se mostrará el identificador (id) de la Sesión, el Usuario, el Juego y la hora a la que se registró esa Sesión.

Mientras que desde el apartado de Juegos, también se pueden ver los detalles, donde aparecerán dos gráficas: una con los objetos usados y otra con el tiempo que ha durado la Sesión. Ambas podrán ser filtradas tanto por Usuario como por Experimento. Esta pantalla se contará con más profundidad en el Capítulo 6.

Todos los usuarios podrán ver los resultados de sus partidas y algunas estadísticas de ámbito global como por ejemplo un top 10 de los jugadores que más rápido han completado un juego, junto a otro *ránking* que contiene a todos los jugadores. Además se pueden ver dos gráficas: una de los objetos usados y otra de los tiempos en los que ha resuelto cierto Juego. Ambas podrán filtrarse primero por Juego y una vez se haya escogido, también por Sesión. A su vez, se le dará la opción de editar sus datos personales (nombre, país, avatar, edad y sexo) y datos de la cuenta (contraseña y correo electrónico).

Por último está el Administrador, el cual es un caso especial ya que este se genera a través de comandos en consola y sirve para gestionar todo lo que se refiere al *back-end*, aunque se le pueden dar distintos permisos. En nuestro caso, le hemos dado permisos para la gestión de las entidades Usuarios, Sesiones, Juegos y Experimentos, entendiendo como gestión las funciones CRUD (*Create, Read, Update, Delete*) de los mismos.

5.1.1. Diagramas de caso de uso del cliente/servidor

En este apartado se mostrarán los diagramas de casos de uso que corresponderán a lo contado en el apartado anterior. Como los nombres y relaciones de los casos de uso son muy claros acerca de lo que hacen o necesitan, hemos optado por hacer diagramas generales en vez de detallar enteros y uno por uno los distintos casos. En primer lugar se encuentra el diagrama de cualquier Usuario, a continuación el diagrama del Administrador y por último, el más complejo", que es el del Investigador.

5.1.1.1. Diagrama de caso de uso de cualquier usuario

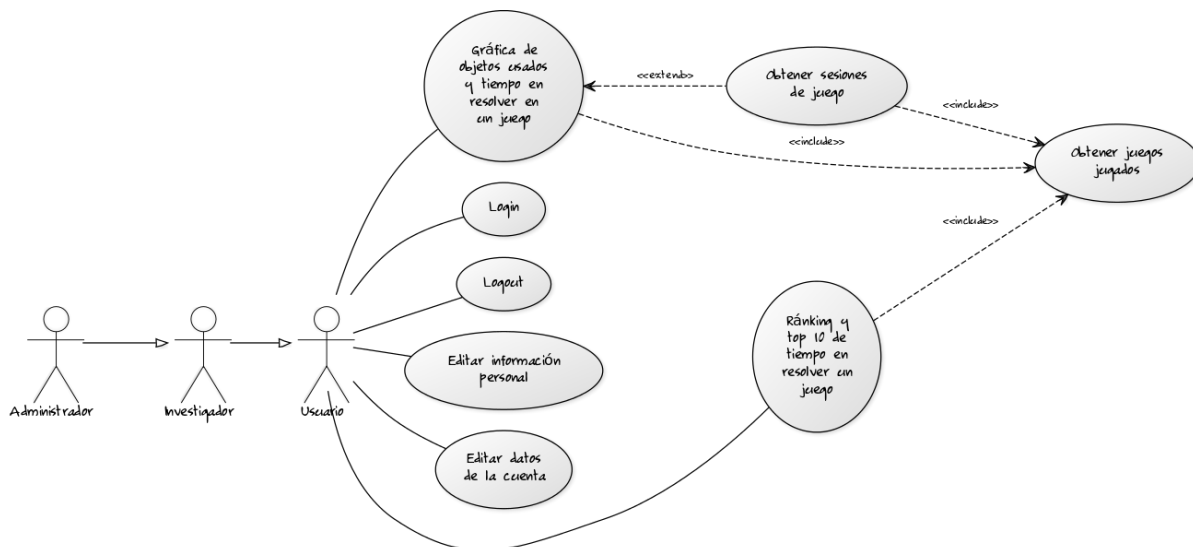


Figura 5.1: Diagrama de caso de uso de cualquier usuario.

5.1.1.2. Diagrama de caso de uso del administrador

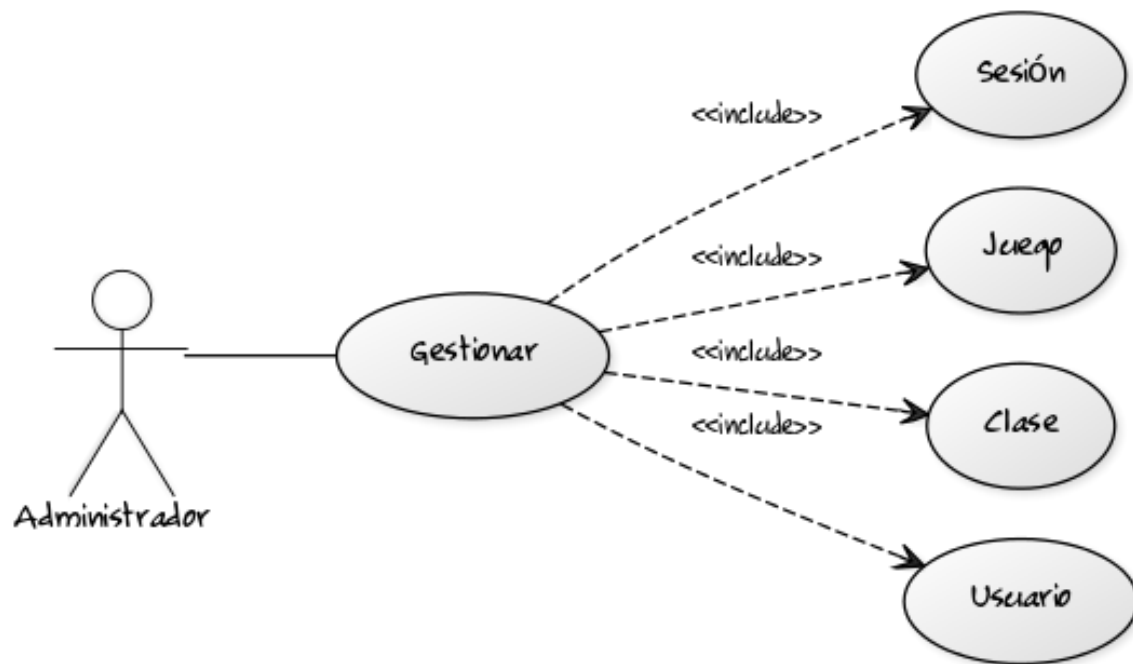


Figura 5.2: Diagrama de caso de uso del administrador.

5.1.1.3. Diagrama de caso de uso del investigador

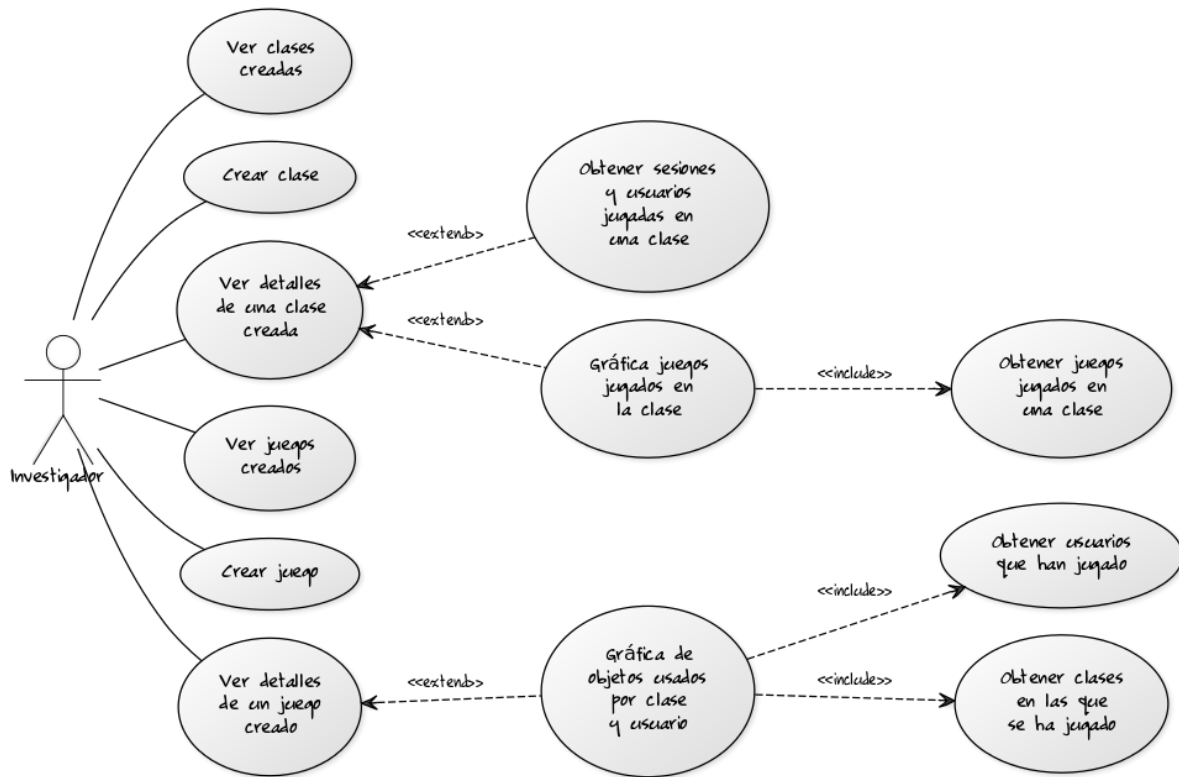


Figura 5.3: Diagrama de caso de uso del investigador.

5.2. Diseño y arquitectura

En esta sección vamos a explicar el diseño y la arquitectura que sigue la aplicación junto a sus diferentes capas.

5.2.1. Diseño de los juegos

Comenzamos diseñando un proyecto en Unity sencillo que consistía en un puzzle para aprender a saber como funciona la tecnología de Oculus Rift junto a la de Leap Motion, con el fin a su vez de recoger datos sobre la sesión de juego. Más tarde hicimos otro juego más complejo y depurado con el que poder recolectar nuevos datos y tener otra aplicación de RV más con la que hacer pruebas.

5.2.2. Diseño de la herramienta

El diagrama de experimentos que sigue nuestra herramienta es el mostrado en la Figura 5.4 el cual fue creado a través del comando

```
python manage.py graph\_models -o models.png polls
```

donde *polls* es el nombre del paquete principal de nuestra aplicación. Este comando es posible hacerse gracias a librería *Graph models* del paquete *django_extensions*. La diferencia que podemos observar respecto a los diagramas de experimento UML es la forma que tiene de representar la agregación, la cual se caracteriza con un punto negro en vez de con el característico rombo.

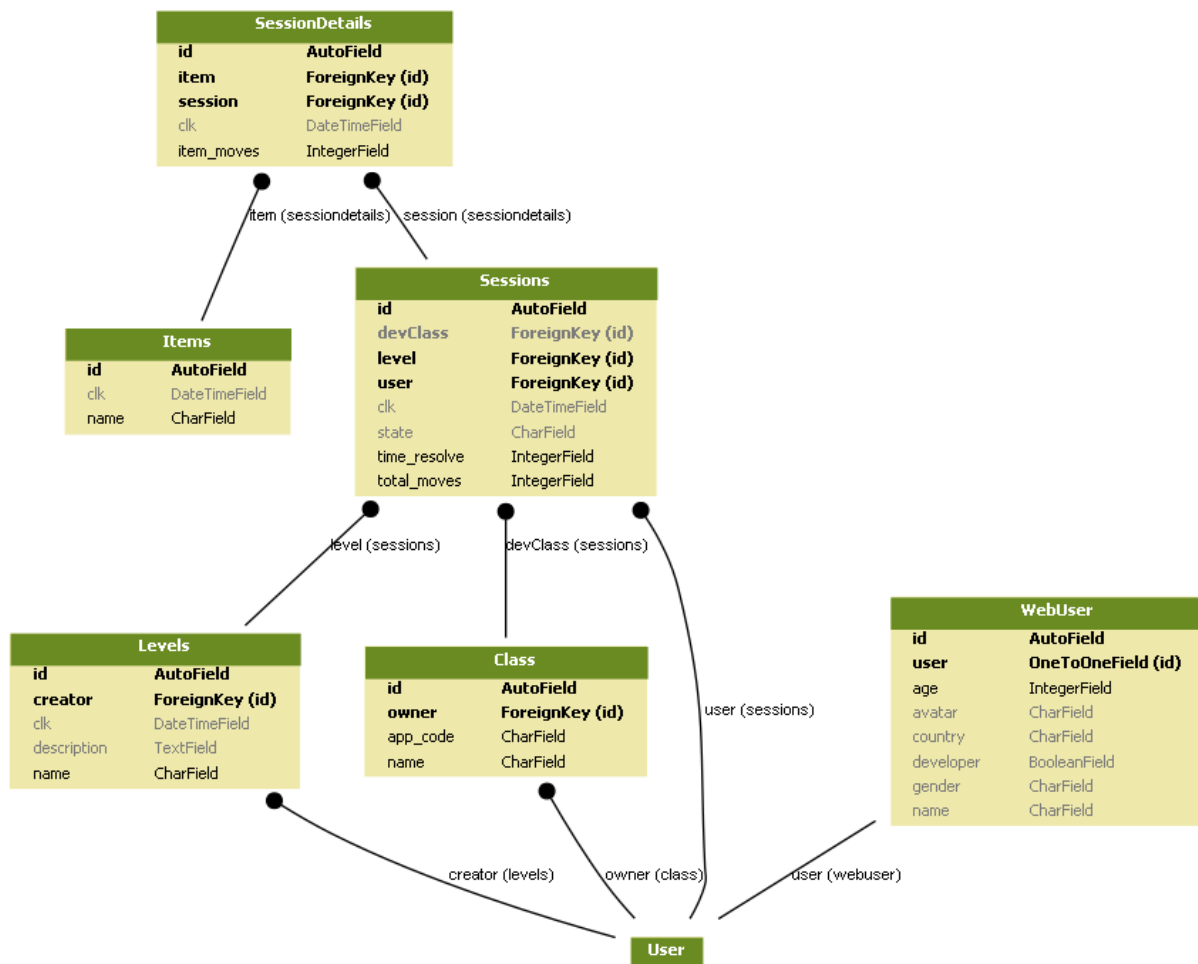


Figura 5.4: Diagrama de clases de la aplicación.

5.2.3. Arquitectura

Nuestra aplicación cuenta con una arquitectura en la que los clientes (*front-end*) solicitan unos servicios al servidor (*back-end*), y este se los proporciona (cliente-servidor). La forma que tenemos de enviar o recibir datos, los cuales serán almacenados o consultados en una base de datos relacional orientado a objetos, se basa en hacer una petición POST al servidor con los datos en formato JSON en caso de querer enviar y GET en caso de querer recibirlos.

Para todo el envío de datos se utilizara el formato JSON¹ a través de una petición HTTP que reciba el servidor.

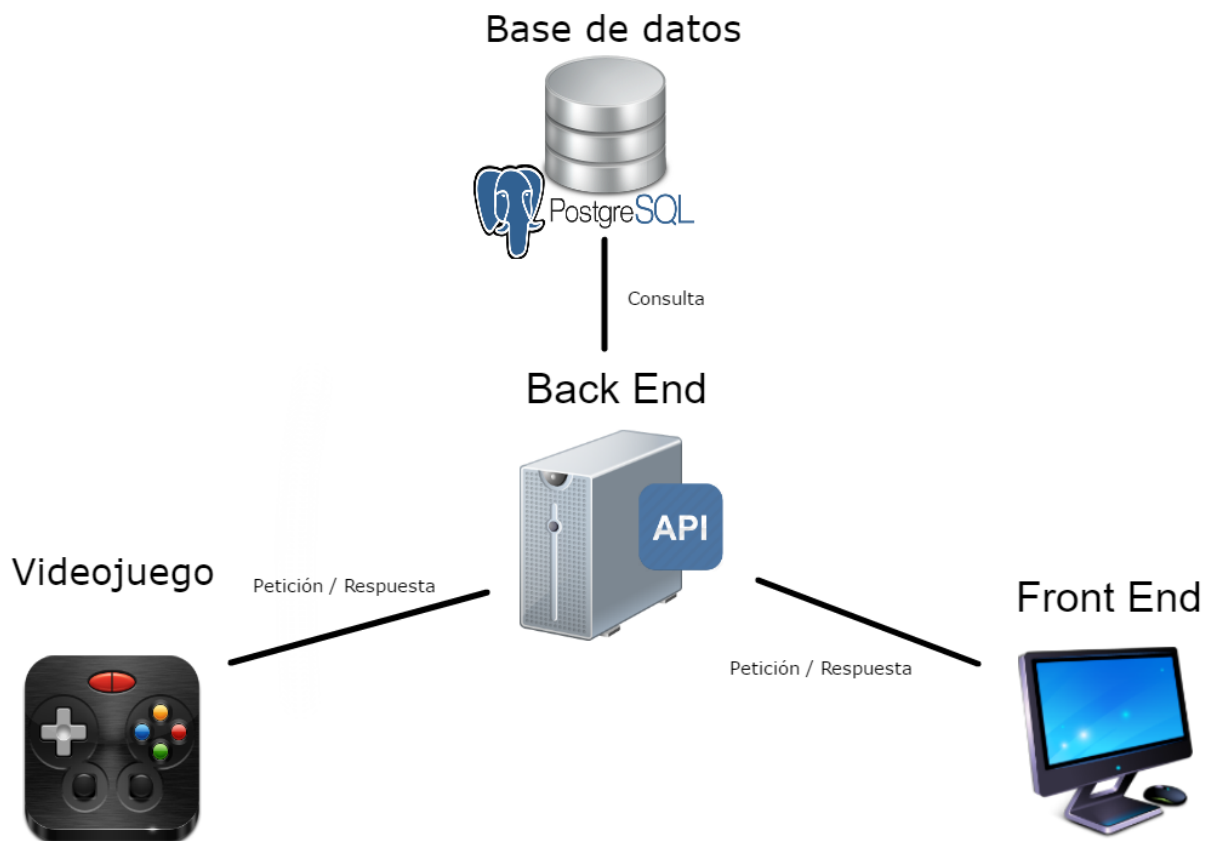


Figura 5.5: Arquitectura del proyecto.

¹JSON: acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

El *front-end* corresponde con la aplicación web que se se encarga de mostrar los datos recogidos en los juegos en forma de gráficas y tablas. Este también cuenta con su propia lógica de negocio para el tratamiento y control de datos, ya que está hecho a través de una tecnología basada en una estructura de Vista, Controlador y Servicios.

Por otro lado, el *back-end* es el encargado de implementar los servicios que hacen falta para la conexión entre el cliente y la base de datos. Además hemos intentado que se ocupe en gran parte del procesado de datos para que el *front-end* tenga la menor carga posible. Un ejemplo de esto es a la hora de enviar datos para la generación de gráficas, las cuales, para rellenarse utilizan listas de objetos con claves específicas, como por ejemplo *value* y *label*. Por ello, toda esta estructura de objetos diccionario la realizamos en el *back-end* ya que el *framework* que usamos lo permite hacer de una manera sencilla.

Para que los clientes puedan solicitar los servicios la parte de acceso a datos ha de estar accesible en Internet. Para ello hemos obtenido un dominio web a través del servicio DNS dinámico (DDNS), el cual permite la actualización en tiempo real de la información sobre un nombre de dominio escogido a través de un servidor de nombres. Este servicio nos proporciona la posibilidad de alojar nuestra aplicación web en uno de nuestros ordenadores sin necesidad de contratar un *hosting*. Concretamente hemos utilizado el proveedor de DNS dinámico No-IP² que permite escoger entre 21 dominios y alojar hasta 5 dominios con su servicio gratuito. Sin embargo, nosotros tenemos dos servidores web uno correspondiente con la parte delantera y otro con la parte trasera y por ello para su continua comunicación hemos tenido que diferenciarles a través de la asignación de puertos.

²No-IP: <http://www.no-ip.com/>

La asignación de puertos ha sido realizada a través del router que hay en la misma casa en la que está el ordenador alojado como servidor. Para el servidor web de AngularJS se escogió el puerto 3000, mientras que para el servidor de Django utilizamos el puerto 8000. De esta manera la redirección queda hecha de la siguiente forma: todo aquel que acceda a nuestro dominio a través del puerto 80, que es el reservado para http, se redireccionará al puerto 3000 el cual se corresponde con Angular, y toda petición que vaya a través del puerto 8000 se redireccionará al mismo 8000 que es el relacionado con Django.

5.3. Implementación

En esta parte de implementación se detallarán las diferentes tecnologías, lenguajes de programación y entornos de desarrollo usados. Además, haremos mención a aquellas tecnologías que teníamos pensado usar pero que no han acabado presentes en nuestra versión final. Por último se hará especial hincapié en el desarrollo de los dos juegos que hemos utilizado para las pruebas de la herramienta y en el desarrollo del sitio web de *Haptic Analytics*.

5.3.1. Lenguajes de programación

5.3.1.1. C#

Lenguaje de programación orientado a objetos, derivado del lenguaje C/C++. Se ha usado este lenguaje para el desarrollo de los *scripts* debido a la amplia comunidad existente en Unity y debido a que Unity basa toda su documentación en C#.

5.3.1.2. Python

Python es un lenguaje de programación que permite adoptar varios estilos, como por ejemplo orientación orientada a objetos, programación imperativa y funcional. A esta característica se le denomina multiparadigma. Además permite de manera sencilla la inserción o uso de extensiones, dando la posibilidad de escribir nuevos módulos fácilmente en C y C++. Se ha utilizado este lenguaje junto al *framework* Django, el cual explicaremos más adelante.

5.3.1.3. HTML5

Lenguaje de marcado para el desarrollo de páginas web. Hemos elegido este lenguaje frente a otros (como AXR o layx) debido a que es el más extendido con diferencia y hemos elegido esta versión debido a que las anteriores versiones están obsoletas.

5.3.1.4. CSS3

Hojas de estilo en cascada. Es un lenguaje de diseño gráfico usado para dar estilos a la página web.

5.3.1.5. Javascript

Lenguaje de programación interpretado orientado a objetos y débilmente tipado. Usamos este lenguaje debido a su sencillez para la gestión del cliente y debido a las diferentes tecnologías que hacen uso de este lenguaje, haciéndolo aún más sencillo de usar.

5.3.2. Tecnologías

5.3.2.1. Unity

Herramienta que ya ha sido comentada en profundidad en el Capítulo 2.

Se ha utilizado tanto para el desarrollo del complemento adicional que va en la parte del cliente de *Haptic Analytics* como para los dos juegos de RV que sirven para probar el sistema.

5.3.2.2. Microsoft Visual Studio

Entorno de desarrollo Integrado desarrollado por Microsoft. Soporta varios lenguajes entre los cuales C++, .NET, Java y C# entre otros.

Posee una versión de pago y otra gratuita llamada Community.

Esta herramienta es la que usamos para la programación en C# de los diferentes scripts necesarios para el sistema y los juegos de demostración.

5.3.2.3. pgAdmin

Herramientas de gestión de base de datos usado para el almacenamientos de los datos recogidos tanto en el sitio web como en las aplicaciones creadas.

5.3.2.4. PyCharm y Sublime Text

Este IDE ha tenido que ser aprendido y posteriormente utilizado para la implementación de la herramienta, tanto para la parte del *back-end* como la del *front-end* una vez se creó la estructura de éste y se juntaron ambas partes. Principalmente ha sido escogido porque está creado específicamente para Python y soporta el desarrollo web con Django. Además, también permite añadir un Sistema de Control de Versiones (VCS) como Git, y una herramienta de depuración.

Sublime Text en cambio es un editor de texto y código fuente, pero que es muy versátil a la hora de programar teniendo varias características como son la multiselección y el multicursor. Fue utilizado principalmente para la creación del *front-end*.

5.3.2.5. Postman

Postman es una herramienta de gestión de API y que permite probar servicios web de manera cómoda, simplemente indicando la URL, el método HTTP y los parámetros a enviar. Además, te permite añadir todo tipo de cabeceras.

Lo más cómodo que tiene es que puedes almacenar estas peticiones en distintas colecciones para su futura reutilización e incluso compartirlas.

5.3.2.6. Django

Django es uno de los framework que usamos para desarrollo web y está, como se ha mencionado antes, escrito en Python. Una de las particularidades que tiene y que por ello se nos hizo fácil trabajar con él, es el patrón de diseño en el que se basa, denominado MVC (Modelo-Vista-Controlador) aunque un poco modificado. Esta modificación tiene como resultado que el llamado Controlador se denomina como Vista, y lo que sería la Vista se llama Plantilla (Template). Sin embargo, esta parte de *front-end* que permite Django nosotros la hemos cambiado por otro servidor de desarrollo web como es Angular.

Respecto a la base de datos, hemos usado la recomendada PostgreSQL, aunque también soporta otras como por ejemplo MySQL y SQLite 3. Esto da entrada a otra de las características que tiene Django y una de las más cómodas que hemos encontrado, que es la abstracción de la base de datos que gracias a su API permite hacer las operaciones CRUD en ella. Además Django permite con gran sencillez crear nuevas tablas y relacionarlas entre ellas a través del modelo de datos que sigue. Una clase o modelo en el *framework* es un registro de una tabla en la base de datos, cuyas filas serán convertidas en instancias. Si se observa la figura 5.4 podemos encontrar los modelos de nuestra aplicación, por lo que cada uno de ellos es una tabla en nuestra base de datos. Igualmente, Django permite al usuario que ejecute sus propias consultas SQL directamente.

En cuanto a la forma que tiene de procesar una petición HTTP, lo primero que hace es crear una instancia `HttpRequest` que permite acceder a todas las propiedades de la petición y diferentes métodos. Una vez está hecha la resolución de la URL, es decir, a través de una url acceder a una vista concreta, se hará todo el procesamiento de datos, operaciones a través de los modelos, y distintos cálculos. A veces el archivo **views.py** es muy extenso y por ello se utiliza otro fichero denominado **helpers.py** unidos al paquete de la aplicación. Por último en el procesamiento de la petición, se generará la respuesta (*response* en inglés) final al usuario.

Finalmente, otra característica o ventaja que tiene Django es su extensa documentación, la cual está separada por la capa del Modelo, la de la Vista y la de las Plantillas entre otras.

El servidor de *Haptic Analytics* será montado principalmente en uno de los ordenadores de los componentes del grupo con la tecnología escogida de Django y una base de datos PostgreSQL que funcionarán como *back-end*.

5.3.2.7. Django Rest Framework

Una extensión que tiene Django es la potente y flexible herramienta para construir Web APIs. En un principio teníamos pensado usarla para el acceso y creación de contenido en nuestra aplicación web, pero la acabamos usándola nada más que para la autenticación de usuarios. Esta decisión se debe a que necesitábamos acceder, crear o actualizar distintos modelos a la vez y con esa estructura se complicaba en la devolución de datos, como por ejemplo se comentó anteriormente, para las gráficas.

5.3.2.8. AngularJS

Este *framework* de JavaScript pensado para la creación y mantenimiento de aplicaciones web de una sola página con estructura MVC, es otra de las tecnologías importantes que hemos usado para el desarrollo de nuestra herramienta, particularmente para la parte de *front-end*. De esta manera tendríamos separadas claramente cada una de las capas de la aplicación y se utilizarían tecnologías no aprendidas en el ámbito universitario.

La hemos elegido principalmente debido a que es un potente y sencillo de utilizar el cual nos proporciona unas herramientas de etiquetado en los archivos HTML para la creación de una aplicación web dinámica. Angular lee estos archivos los cuales contienen etiquetas adicionales y se relacionan obedeciendo ciertas directivas con los valores de variables de JavaScript. Estos valores se pueden modificar de forma manual o dinámica a través de recursos JSON.

Este framework por lo tanto adapta y amplía el HTML tradicional a uno con mejor contenido dinámico gracias a la tecnología llamada *data binding*, que consiste en vincular los elementos de la interfaz de usuario con los datos que recibimos de nuestro *back-end*. Esto permite que no tengamos que preocuparnos por la actualización de las vistas cuando los datos cambian.

Las directivas anteriormente mencionadas que tiene Angular y las cuales se colocan sobre elementos del DOM se dividen en dos tipos: directivas nativas y directivas propias. Nosotros explicaremos a continuación solamente las que hemos utilizado que son las nativas o existentes, y entre ellas, las que consideramos más importantes.

- **ngApp (ng-app)**: la directiva más importante y la encargada de arrancar una aplicación Angular. Ésta indica el elemento principal y se debe colocar en el elemento que se quiera como raíz de la aplicación. En nuestro caso está en el .html principal de la siguiente manera:

```
<body class='mainApp'ng-app='app'ng-controller='MainController'>...</body>
```

La cual contiene a su vez la siguiente directiva, ng-controller.

- **ngController (ng-controller)**: es la directiva que permitirá indicarle a la vista donde trabajará nuestro controlador y vincularlo de tal manera que podamos acceder a sus atributos y funciones. Nosotros tenemos cuatro controladores:
 1. Main Controller: se encarga del estado general de la aplicación y guardar algunas variables globales.
 2. User Controller: contiene toda la información relacionada con el usuario.
 3. Statistic Controller: almacena la información acerca de las estadísticas del usuario en los juegos y comparada con otros usuarios. Esta información será utilizada para la generación de gráficas.

4. Class Controller: igual que Statistic Controller pero con las experimentos y juegos.
- **ngModel (ng-model)**: esta marca representa al dato y permite obtener la información ingresada manual o dinámicamente. Se usa principalmente para los elementos del DOM, como por ejemplo input, select o textarea. Un ejemplo de nuestra aplicación:

```
<select class="form-control" id="session" name="session"
  ng-model="selectedSession"
  ng-init="selectedSession='' "
  ng-change="getPieChartData(selectedLevel,selectedSession);">
  <option value="">All</option>
  <option ng-repeat="session in sessionList" value="{{session.id}}">{{session.name}}</option>
</select>
```

Figura 5.6: Ejemplo de uso de la directiva ng-model.

En este caso *selectedSession* será la variable que corresponde con la directiva ng-model. A su vez usamos dos directivas más, ng-init que sirve para inicializar, en la que *selectedSession* se inicializará vacío y ng-change, cuya funcionalidad es llamar hacer una llamada al *back-end* dependiendo del valor que contenga la variable. Por último, está la directiva ng-repeat que ejercerá de bucle para generar las opciones a escoger del select.

Además, AngularJS cuenta con una amplia comunidad que desarrollan nuevos módulos como lo es Satellizer, la cual nos permite hacer la autenticación del usuario basada en un token. Este lo recibimos haciendo una llamada al framework Django REST. Otro módulo que usamos para AngularJS es angular-translate con el que se hace de manera sencilla una traducción del contenido.

5.3.2.9. Satellizer

Es un módulo de AngularJS desarrollada por Sahat Yalkabov ³, que simplifica la autenticación basado en Token. Soporta la autenticación de cualquier sistema basado en el estándar OAuth 1.0 ó 2.0. En nuestro caso, lo hemos usado para nuestro sistema de autenticación que usa el estándar JWT ⁴

5.3.2.10. Bootstrap

Otro de los framework que hemos utilizado, esta vez para el aspecto de la página web, es Bootstrap. La razón por la que hemos escogido utilizarlo es que proporciona gran cantidad de estilos y contenidos, además de ser el que hemos usado la mayoría de las veces y nos resulta muy sencilla su práctica.

5.3.2.11. FusionChart

Este *plugin* desarrollado en Javascript y que se puede implementar perfectamente con AngularJS a través de las distintas directivas explicadas y mencionadas anteriormente nos permite visualizar los datos recogidos por el *back-end*. Una de las mejores cosas que tiene es la amplitud de gráficas y la gran cantidad de ejemplos que presentan en su página web. Además, una de las cosas que presumen es el acceso que puedes tener a su código, junto a la modificación de eventos, atributos, anotaciones, temas, etc. Para hacer mejor aún este *plugin* permiten la exportación de las gráficas en distintos formatos, como por ejemplo PNG, JPG, PDF, SVG y XLS.

³Sahat Yalkabov: <http://sahatyalkabov.com/>

⁴JWT (JSON Web Token): Es un estandar abierto basado en JSON para la creación de Token de acceso. <https://jwt.io/>.

5.3.2.12. Elasticsearch y Kibana

Tanto Elasticsearch como Kibana son herramientas que teníamos pensado utilizar para nuestra herramienta debido a que permiten el almacenaje de datos en formato JSON y la posterior visualización de ellos a través de gráficas. Aunque ambas tecnologías estén incorporadas en nuestro proyecto y listas para ser usadas potencialmente no hemos terminado de llegar a implementarlas, ya que la manera que tiene de hacer las gráficas con Kibana es muy manual, o al menos nosotros no descubrimos como generarlas automáticamente y a nuestro gusto, seleccionando los campos que se desean mostrar. Es por ello que no haremos mucho más hincapié en estas dos tecnologías. Sin embargo, cabe mencionar su destacado motor de búsqueda que es Elasticsearch, el cual permite encontrar todo tipo de documentos y su arquitectura, que actúa como un coordinador de operaciones entre los distintos índices y fragmentos.

5.3.3. Desarrollo de los juegos

En esta sección contaremos el desarrollo de los dos juegos que hemos creado para poner a prueba nuestra herramienta. El primero de ellos, *Cube Board*, aunque tiene solución, no termina de explotar todas las posibilidades de la RV ni tiene una jugabilidad muy rica. Por ello acabó creándose el segundo videojuego de RV, el cual denominamos *Escape Room*, que tiene muchos más aspectos interesantes para analizar.

5.3.3.1. Cube Board

Este juego ha tenido dos versiones a lo largo de su desarrollo en las que variaba el hardware con el que interactuar con los distintos elementos.

Al no tener disponible todo el tiempo las Oculus Rift, la primera versión presentada utilizaba el control de Leap Motion para la interacción con los objetos mientras que para simular el movimiento que tendría la cámara si usásemos las Oculus Rift se hacía con el ratón. Sin embargo en la segunda versión ya disponíamos de las gafas de realidad virtual a tiempo completo, pero no todos los componentes del grupo tenían unas. Por ello se implementó una manera un modo depuración con el que el juego podía ser jugado solo con el ratón, para que los demás miembros del equipo que no tuvieran en ese momento las Oculus Rift o Leap Motion pudieran generar datos.

A continuación explicaremos el diseño que han seguido las siguientes versiones:

■ Versión 1 - Diseño básico del nivel y su jugabilidad

El juego consiste en un tablero con dos caras, uno en el que podíamos mover libremente las piezas del tablero (parte juego) y la otra con los símbolos fijos (parte solución). El objetivo del juego era hacer coincidir el elemento de la parte juego con su contrario en la parte solución y que así desaparecerían dejando un hueco libre en ambos lados. Estos elementos se generan de manera aleatoria en ambas partes.

Para poder ver la parte solución, se disponía de un botón de color rojo en la parte derecha del tablero, el cual hacía rotarlo 180 grados y dejaba ver la solución unos instantes para luego volver a la parte juego.

El procedimiento concretamente sería:

1. Pasar la mano por encima del botón rojo para poder observar la parte solución
2. El tablero rotaría 180 grados en el eje Y
3. Veríamos entonces que en la parte inferior derecha hay una pieza de tipo fuego
4. El tablero en unos instantes volvería a la rotación original
5. Moveríamos una pieza de tipo agua en la parte inferior izquierda
6. Al estar en la parte solución el elemento contrario, los dos desaparecerían

Debido a un error de cálculos, el juego no se desarrolló con la escala correcta, por lo que tuvimos que adaptar la posición relativa de las manos a la posición a escala en el mundo de la escena del juego. Esto nos llevó a que cada mano interactuara con las piezas o el botón de rotación de tablero a través de un *RayCast*, el cual consiste en un rayo que va desde la posición de la mano hasta el objeto con el fin de acceder a su experimento y posteriormente poder interactuar con él.

■ **Versión 2 - Ampliación de los distintos modos de juego**

Se añadió una variante del juego en el que la parte juego tenía piezas ocultas, si se mostraba la mano izquierda que tenía un efecto fuego y pasaba por encima de estas piezas, se mostraba su verdadero elemento durante unos instantes en los cuales el jugador tendría que memorizar su elemento.

Contaba con tutoriales para saber cómo funcionaba el juego.

Se añadió otro modo de juego en el que cada vez que se hacía la combinación de un elemento, el elemento movido aparecía en una torre de cubos de ese elemento, si el jugador hacía demasiadas eliminaciones del mismo elemento seguidas se caía y el jugador perdía. Había cubos en la parte izquierda del nivel que el jugador podía mover libremente, sin estar fijados a un tablero, porque debido a la aleatoriedad del nivel podía pasar que solo se pudiera combinar un elemento, se podía coger estos cubos para equilibrar las torres y así poder superar el nivel.

5.3.3.2. Escape Room

- **Versión 1 - Creación de primer nivel, modo ratón primera versión VrTracker** Debido a la pobre jugabilidad que tenía *Cube Board*, donde el jugador ni siquiera podía moverse y navegar por un entorno virtual, decidimos crear otro juego en el que, con la experiencia obtenida al desarrollar el anterior, hubiese una mayor amplitud de características a explorar. Esto se haría con un modelo de proceso iterativo e incremental, cerrando versiones cada pocas semanas.

Al ser un proyecto en RV en el que se podía hacer uso de las manos gracias a Leap Motion, un tipo de juego que podría encajar perfectamente con el uso de estas tecnologías sería uno de género “sala de escape” (en inglés *Escape Room*). No tendría mucho movimiento y los usuarios no tendrían porque sufrir tanto las desventajas de la RV, como por ejemplo mareos y náuseas. Además, el uso de Leap Motion podría sernos de gran utilidad para implementar diversas acciones. El objetivo de estos juegos es que el jugador salga de un sitio en el que se encuentra encerrado resolviendo diferentes puzzles o problemas.

Al empezar el juego desde cero y con más conocimientos que teníamos al empezar *Cube Board*, éste pudo realizarse con una escala correcta en los objetos y en el escenario, y dividiendo los objetos interactivables en distintos experimentos para ser más sencillo el envío de datos a posteriori. Esta división es la siguiente:

- **Mover libremente:** una piedra, un vaso o cualquier objeto que podríamos recoger y lanzar
- **Mover tablero:** una pieza igual que en el *Cube Board* que la pudieras mover pero únicamente dentro de un tablero
- **Menú botón:** un botón con el que el usuario pudiera interactuar. Esto pierde sentido al realizarse el *login* con el teclado y fuera de RV.
- **Botón:** una pieza o botón que el usuario pudiera presionar con la mano izquierda para provocar alguna reacción.

Movimiento del jugador

En esta versión, el jugador se movía cuando miraba cierto tiempo a sitios concretos predispuestos en el nivel. El escenario y los personajes que aparecen son propios de una aventura onírica y extraña.

Descripción del nivel

El nivel se compone de dos salas, unidas por un pasillo.

1. *Primera sala.* Comenzamos en una celda de estilo fantástico-medieval, en la que la idea inicial era que estuviera muy oscura con la única iluminación de unas antorchas externas a ella. Para salir tendríamos que coger una de las piedras que están en la celda y lanzarla a la antorcha y una vez que ésta cayera pudiéramos ver. En consecuencia, se produce un cambio de ángulo en la proyección de la luz que descubre unas baldosas en la pared que sobresalen, las cuales debemos presionar sin ningún orden preciso. Una vez presionadas cuatro veces, se deja ver una trampilla dándonos la posibilidad de pasar a la siguiente sala cruzando antes el pasillo que las une.
2. *Segunda sala.* En esta segunda sala, nos encontramos en una habitación que contiene un piano que solo tiene 5 teclas y en cada una un número, encima del piano nos encontramos una nota que nos da la pista del puzle “Si hay magia en este mundo, está contenida en el agua”, un hombre sentado en una silla cerca de una mesa que contiene una nota vacía, un vaso derramado y una botella, una cama y una mesita de noche. También en la habitación cerca de la mesa se encuentra una gotera, lo que tenemos que hacer es coger la nota vacía y situar debajo de la gotera, al mojarse nos desvelará la clave, que tenemos que introducir en el piano, simplemente tocando las teclas en el orden correcto. Una vez realizado esto el piano se ocultará hacia abajo y podremos acceder a la siguiente fase del nivel.



Figura 5.7: Detalle de la segunda sala

Creación del modulo VrTracker para el registro de movimiento de objetos

Empezamos haciendo un *script* simple que buscaba los objetos dinámicos y les incorporaba otro *script* a cada uno para llevar el seguimiento de cuantas veces se movían durante la sesión de juego. El objetivo es que al final de ésta, la información de los objetos se agruparan y se enviaran aquellos que hubieran tenido al menos un movimiento y con un tiempo de vida mayor que un segundo. Esto se hizo para evitar objetos recurrentes como por ejemplo una gota de agua que cae del techo.

En la primera versión del modulo no teníamos creados los modelos en el servidor para almacenar la información por lo que íbamos guardando los datos de los objetos en un JSON.

- **Versión 2 - Adaptación a Oculus Rift + Leap motion, primeras pruebas de registro de jugadores y envío de información al servidor**

Debido al cambio de realizar las pruebas con Oculus Rift tuvimos que hacer numerosos cambios en las dos salas, en la primera sobre todo porque la sensación de oscuridad desaparece y en la segunda debido a las limitaciones de tener un dispositivo con tantas conexiones mediante cable.

Movimiento del jugador

El sistema de movimiento anterior dio numerosos fallos así que al final terminamos por hacer que el movimiento se produjera si el jugador mostraba la mano izquierda y cerraba el puño, entonces se movería en la dirección que estuviera mirando sin limitaciones. Un método menos inmersivo pero eficaz.

Cambios en las salas del nivel

1. *Primera sala:* debido al cambio de dispositivo solo teníamos que pulsar las distintas baldosas que sobresalían al poder verlas con tanta claridad.
2. *Segunda sala:* en la segunda sala y el pasillo reorientamos el nivel para que principalmente el jugador mirara en una dirección y tuviera que darse la vuelta el menor número de veces.

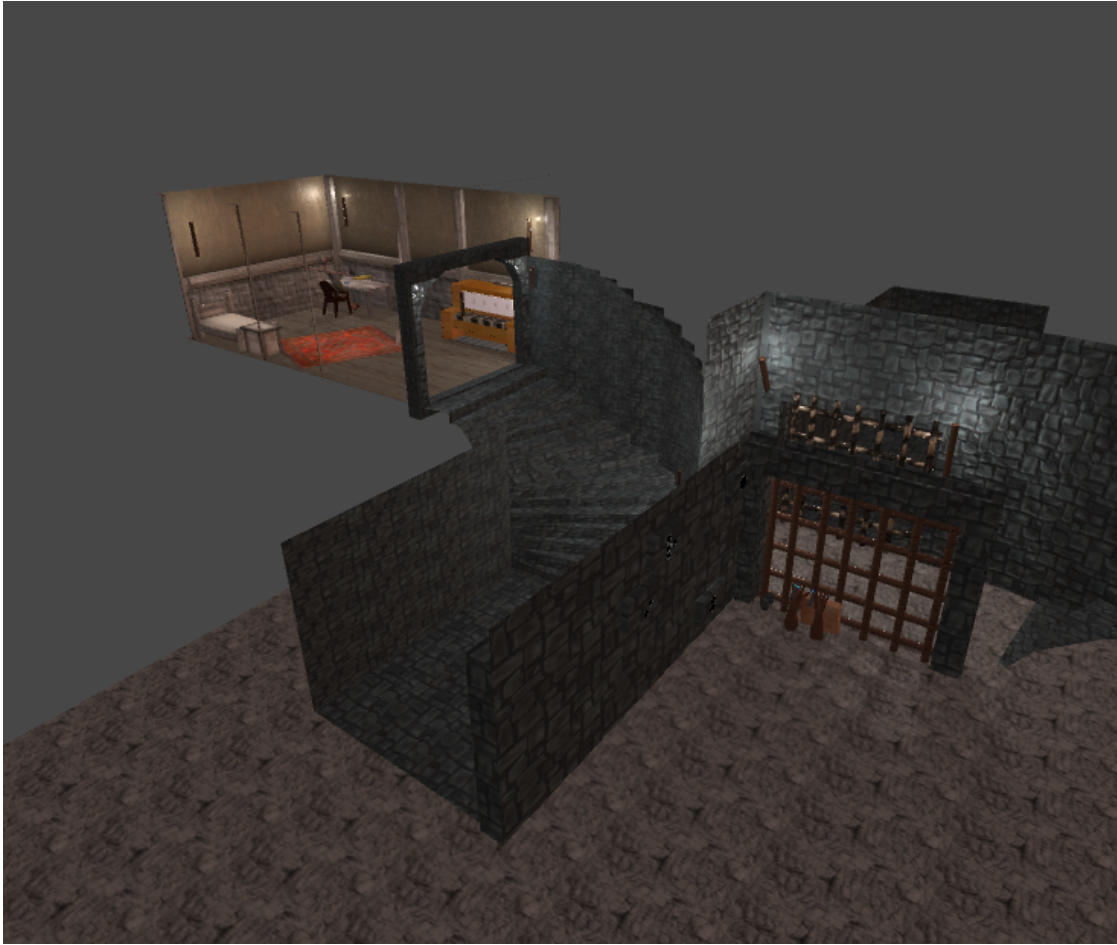


Figura 5.8: Detalle de nivel del juego

Evolución del modulo VrTracker para el registro de movimiento de objetos

Hicimos una escena aparte en la que los jugadores se registraban para poder pasar una vez completo el registro al juego *Escape Room*. El modulo contó con un filtro por nombres concretos y otro por objetos que contengan palabras descritas por el desarrollador, muy útil para eliminar objetos innecesarios con nombres parecidos.

El módulo desarrollado sería una versión básica, pensada para que sirviese a videojuegos tanto tradicionales como de VR, dedicándose a medir el uso de los objetos normales, para que fuera más genérico. Además, ya se crearon los modelos necesarios en el servidor para poder almacenar los datos junto a las llamadas correspondientes y comprobamos el buen funcionamiento del envío de información.

- **Versión 3 - Mejora de la primera sala, optimización de modulo VrTracker y prueba en un juego ajeno**

Mejora de la primera sala

A causa de la escasa sensación de oscuridad que teníamos con Oculus Rift cambiamos la idea de pulsar todas las baldosas por un puzle a resolver. Este puzle consiste en una tablilla con un alfabeto en Braille, y las baldosas que sobresalen contienen la simbología de las letras 'A', 'B', 'R' e 'I', para que el objetivo del jugador sea construir la palabra 'ABRIR' mediante la pulsación en orden de los botones.

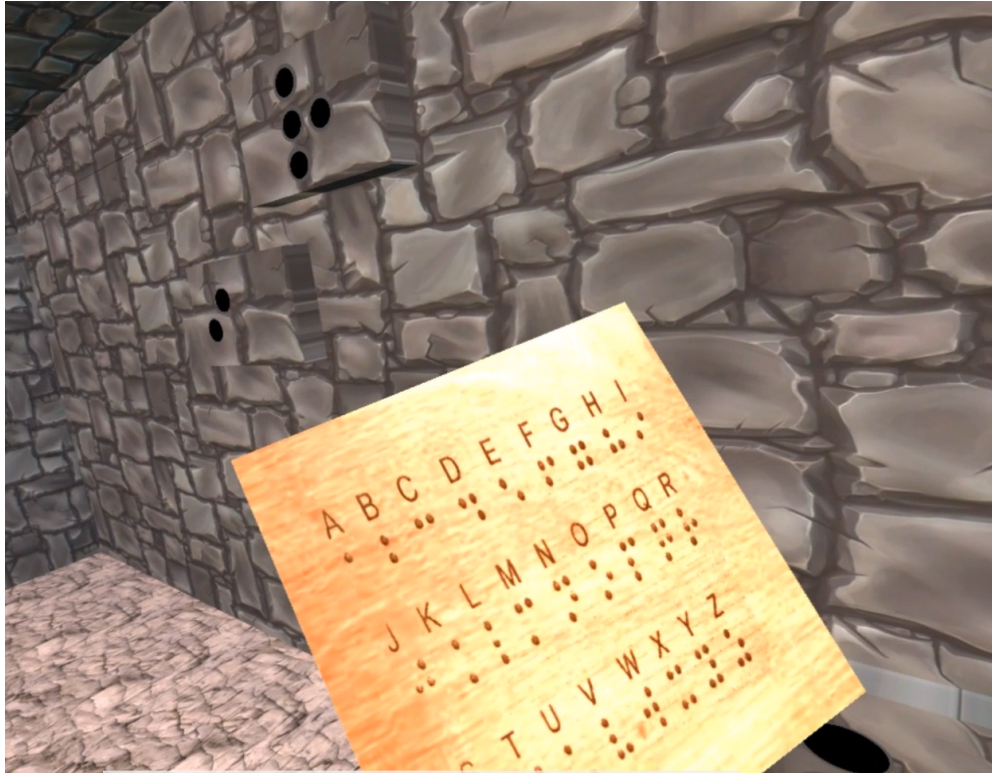


Figura 5.9: Detalle de la primera sala, con el enigma del braille.

Optimización del modulo VrTracker y prueba en un juego ajeno

Para comprobar que verdaderamente podría funcionar en un juego ajeno no desarrollado por nosotros y probar lo genérico que era, lo probamos en un proyecto completo de los que vienen como ejemplo en el propio entorno de desarrollo Unity. También para comprobar si era tan fácil de adaptar, hicimos numerosos cambios para que se quedara en un simple paquete de Unity el cual consta de una escena prefabricada de registro de jugadores, y un solo *script* que se debe añadir al GameController o controlador de juego creado por el desarrollador en el juego.

5.3.4. Desarrollo del sitio web

En esta sección hablaremos del desarrollo del sitio web que tiene como finalidad el mostrar a los investigadores los resultados recolectados de los experimentos.

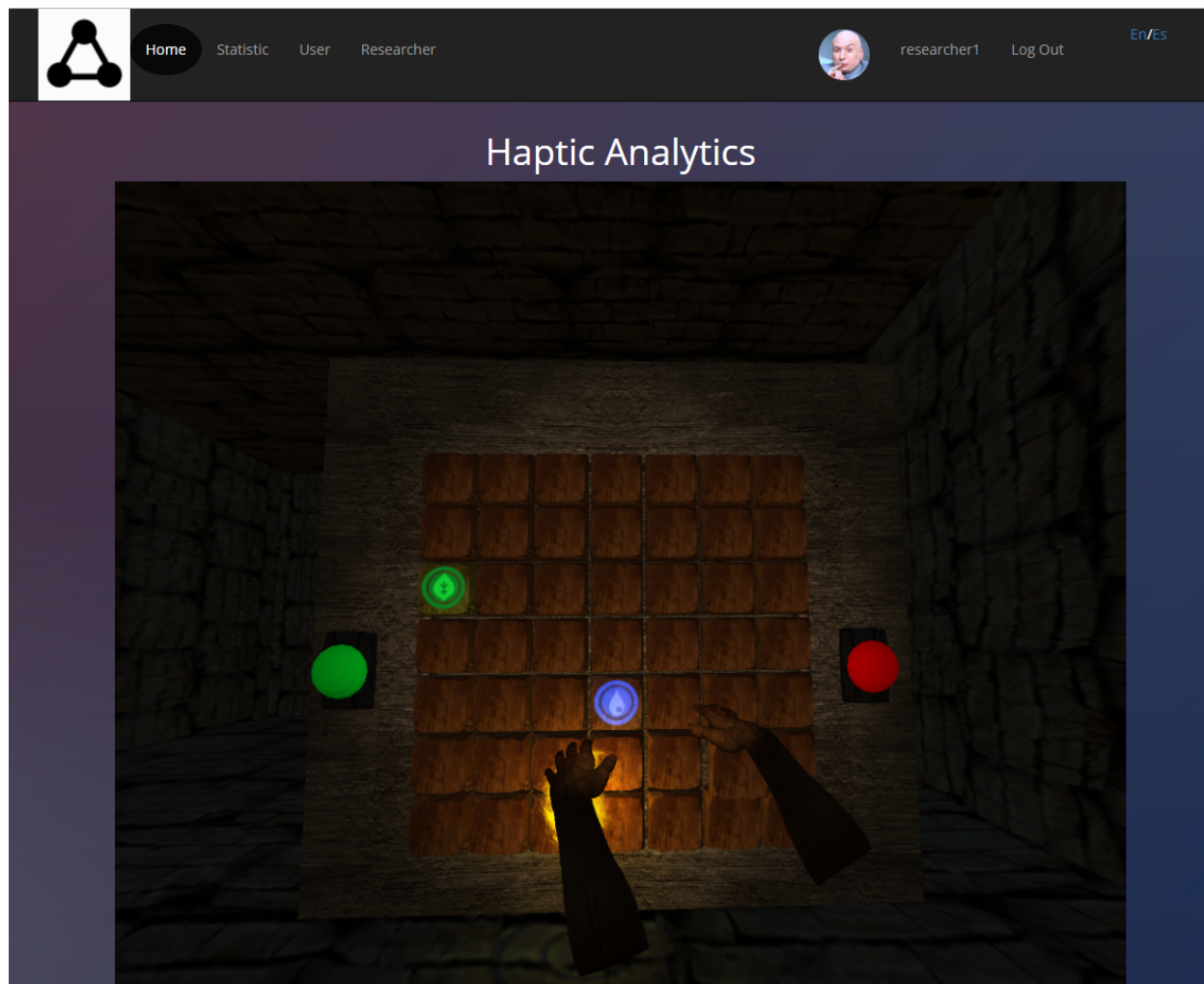


Figura 5.10: Página principal del sitio web.

Como se puede observar en la figura 5.10, en la parte superior contamos con un menú que está formado por los siguientes elementos:

1. Logo de la aplicación web
2. Mapa de las principales secciones
3. Apartado de usuarios
4. Idioma de la aplicación web

5.3.4.1. Logo

Aquí es donde se muestra el logo del sitio web. Actualmente estamos usando el logo de Narratech Laboratories. Si se hace clic sobre el logo, redirige hacia la página principal de la web.

5.3.4.2. Mapa de la aplicación web

Este mapa contiene las principales secciones de la aplicaciones web y las cuales se explicarán a continuación, que son:

1. Home
2. Estadísticas
3. Usuario
4. Investigador

5.3.4.3. Home

En esta sección se muestra la portada de la aplicación web. Contiene el título de este trabajo y una imagen del juego *Cube Board* a modo de portada.

5.3.4.4. Estadísticas

Esta sección es donde se muestran las gráficas y tablas individuales del usuario. Dispone de un menú lateral que contiene las subsecciones:

- Ránking
- Gráficas

En Ránking podemos encontrar estadísticas en forma de tablas. Podemos ordenar por una columna en concreta haciendo clic en el título de esa columna.

En Gráficas podemos encontrar estadísticas en forma de gráficas circulares o de puntos. Más adelante explicaremos esta sección más a fondo.

5.3.4.5. Usuario

En esta sección puede cambiar la información de su perfil (figura 5.11) y los datos de acceso a la cuenta.

Dispone de un menú lateral que contiene las subsecciones:

- General
- Contraseña y correo

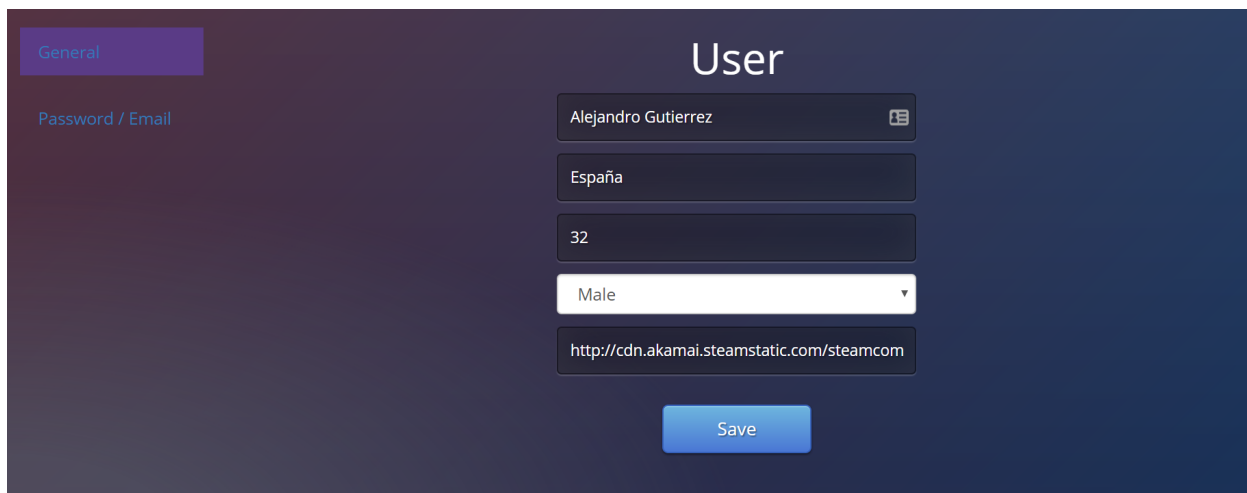


Figura 5.11: pantalla para cambiar el perfil del usuario.

La información del perfil que el usuario puede cambiar son:

- Nombre
- País
- Edad
- Género
- Enlace del avatar

Cuando el usuario no tiene ningún avatar, se muestra un avatar por defecto (figura 5.11 a).

Si hay algún problema al intentar obtener el enlace del avatar, se muestra un avatar de error por defecto (figura 5.11 b).

A destacar que el enlace del avatar tiene que ser completo (incluido "http://").

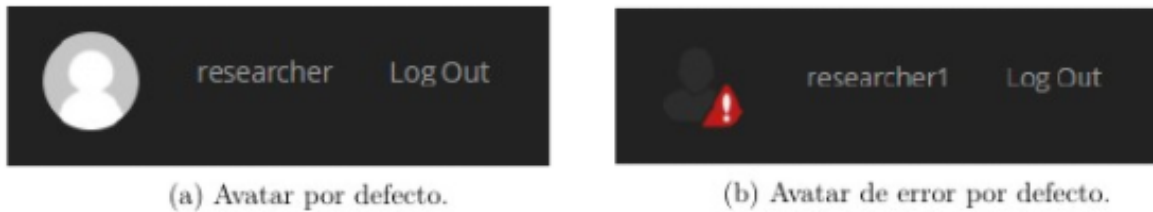


Figura 5.12: Avatar por defecto.

La información de acceso que el usuario puede cambiar son:

- Correo electrónico
- Contraseña

5.3.4.6. Investigador

Por último, tenemos la sección en donde se muestran las estadísticas en forma de gráficas de cada experimento y de cada juego creado por el investigador.

Dispone de un menú lateral que contiene las subsecciones:

- Experimentos
- Juegos

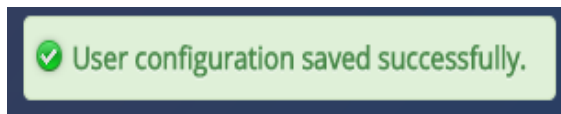
Más adelante explicaremos esta sección en profundidad.

5.3.4.7. Otras características

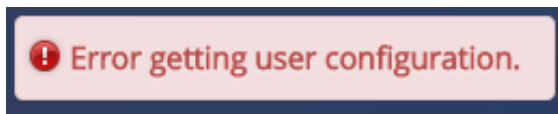
Aquí detallaremos otras características que tiene la aplicación web que no tiene cabida en otras secciones.

- **Notificaciones:** la aplicación web cuenta con un sistema de notificaciones para gestionar los diferentes mensajes de respuesta al usuario, tales como cambios guardados correctamente, error al guardar cambios, etc.

Las notificaciones aparece en la esquina superior derecha, justo debajo del menú superior, y desaparece tras 5 segundos o haciendo clic sobre la notificación.



(a) Notificación de correcto.



(b) Notificación de error.

Figura 5.13: Sistema de notificaciones.

Este sistema de notificaciones se ha implementado usando la librería Notify.

- **Traducciones:** la aplicación web también cuenta con dos idiomas, el inglés y el español, siendo el inglés el lenguaje por defecto.

This screenshot shows the 'User' profile page in English. The top navigation bar includes 'Home', 'Statistic', 'User' (highlighted), and 'Researcher'. On the right, there is a user profile icon, the text 'researcher', a 'Log Out' button, and a language toggle 'En/Es'. The main content area has a 'General' tab selected on the left. The form fields are labeled 'Name', 'Country', 'Age', '-- Gender --' (a dropdown menu), and 'Avatar link'. A 'Save' button is at the bottom.

(a) Web en inglés.

This screenshot shows the 'Usuario' profile page in Spanish. The top navigation bar includes 'Home', 'Estadísticas', 'Usuario' (highlighted), and 'Investigador'. On the right, there is a user profile icon, the text 'researcher', a 'Desconectar' button, and a language toggle 'En/Es'. The main content area has a 'General' tab selected on the left. The form fields are labeled 'Nombre', 'Ciudad', 'Edad', '-- Género --' (a dropdown menu), and 'Enlace de avatar'. A 'Guardar' button is at the bottom.

(b) Web en español.

Figura 5.14: Web en diferentes idiomas.

Se puede cambiar entre los dos lenguajes haciendo clic sobre el lenguaje deseado en la esquina superior derecha del menú superior.

Capítulo 6

Pruebas y resultados

En este capítulo resumimos las pruebas y comprobaciones que hemos ido realizando a lo largo de todo el proceso de desarrollo, explicando cómo es exactamente la funcionalidad que hemos llegado no sólo a implementar sino a validar en pleno funcionamiento. Comenzando desde la implantación de nuestra herramienta *Haptic Analytics* en un juego del estilo a los dos juegos de RV que hemos desarrollado nosotros, hasta la consulta y visualización de los datos de una Sesión recién terminada en el sitio web que hemos creado.

6.1. Implantación y configuración

Para la implantación de nuestra herramienta en cualquier juego hecho en Unity hace falta descargarse el paquete. Una vez importado el paquete, obtenemos en nuestra carpeta de *Assets* un *prefab* de los objetos que tendremos que añadir a la escena que queremos que sea la principal, en la que se registran e inician sesión los jugadores.

Por otra parte tenemos el objeto *VrTracker* el cual se pone en la escena a analizar, y tiene que ser llamado desde el juego mediante un función. Ésta función tiene que indicar si ha ganado o perdido y el tiempo de juegos. Tras la llamada, se envía los datos al servidor.

- Creación de la escena de login.

Creamos una escena vacía e incorporamos el *prefab* de login que tenemos en la carpeta *VrTracker*, y la configuramos a nuestra necesidad en el *script*.

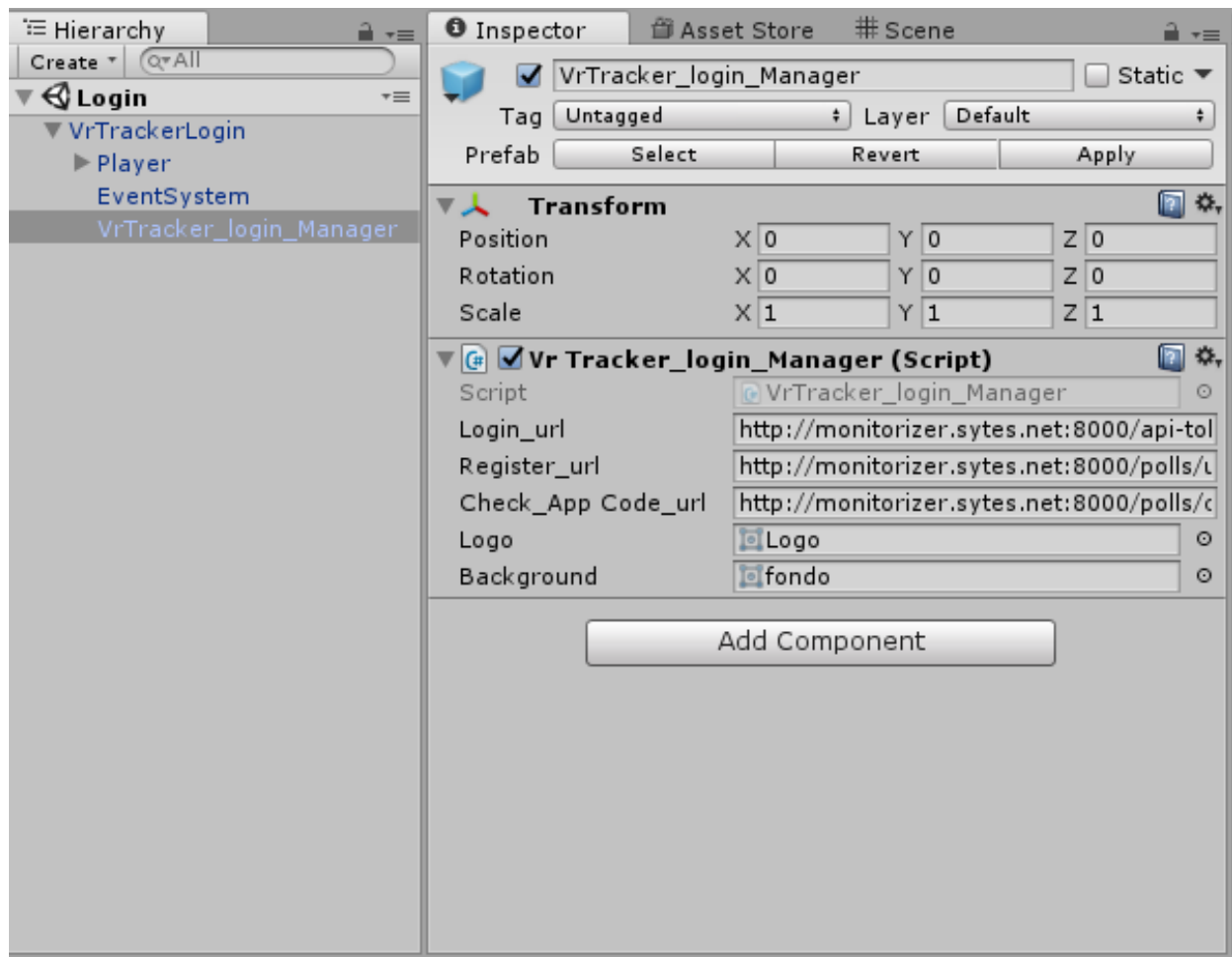


Figura 6.1: Figura de datos de login.

- Incorporación del módulo *VrTracker*

Tenemos que añadir el *script VrTracker* al objeto que tengamos con la etiqueta *Game-Controller* de la escena de Unity. Este *script* en principio deberá ser llamado cuando el jugador termine la partida con la siguiente llamada, aunque realmente se puede utilizar en cualquier momento.

```
this.GetComponent<Vr_Tracker>().send_info("win",Time.time);
```

En este caso se ha llamado desde el mismo objeto y una vez el jugador ha ganado. El segundo parámetro, *Time.time*, se utiliza para enviar el tiempo que ha transcurrido entre el inicio (T_i) y el final de la sesión (T_f).

También se ha añadido la funcionalidad de añadir un objeto y movimiento calculado por nosotros mismos a el envío de información mediante la siguiente llamada:

```
public void sendMoreInfo(string name, int movimiento);
```

Finalmente se hicieron pruebas de integración con una aplicación de terceros, un juego totalmente ajeno a nuestro proyecto descargando de la Asset Store de Unity, una plantilla de juego llamado TopDown_AI.

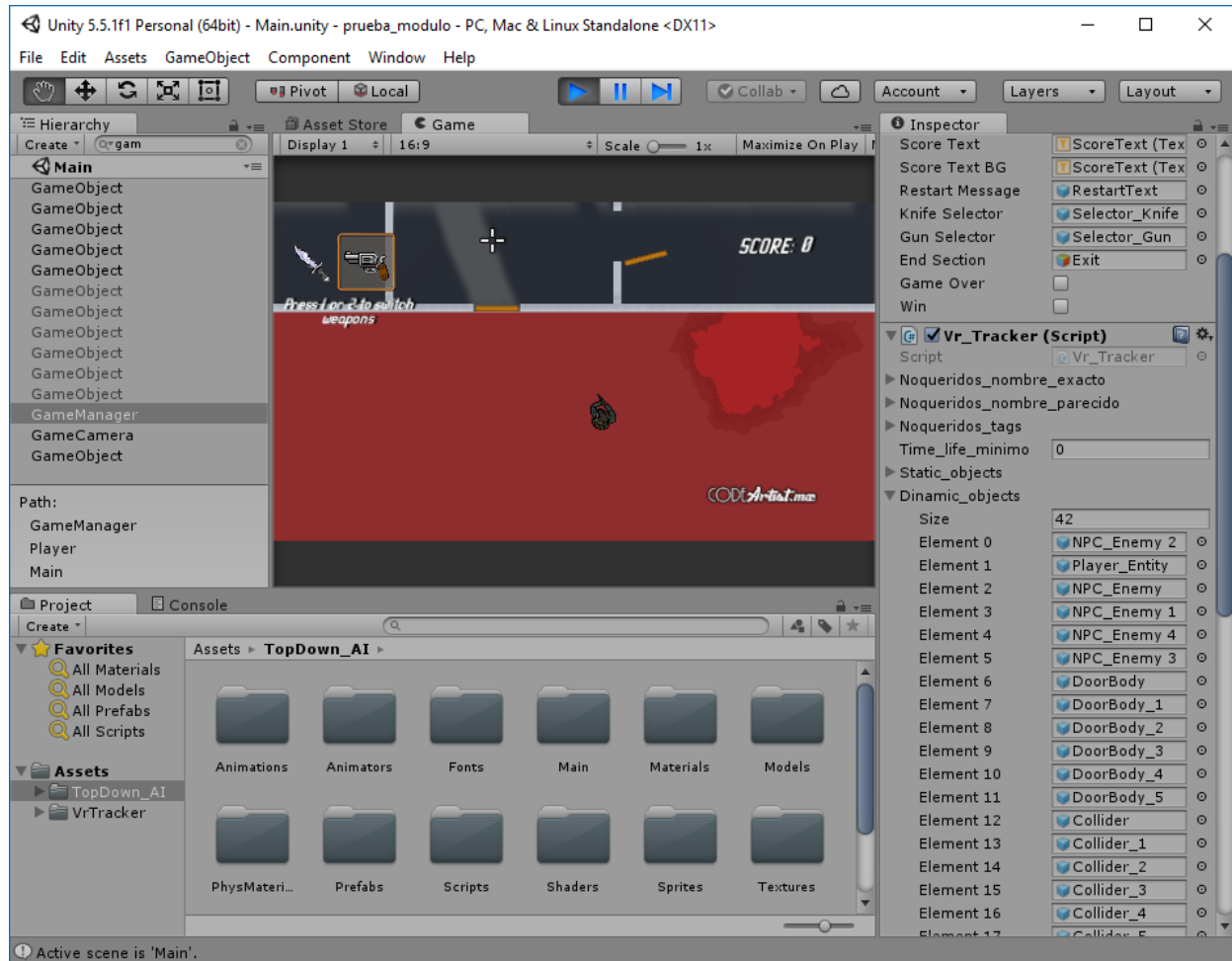


Figura 6.2: Captura del modulo en funcionamiento dentro de TopDown_AI.

- Formato JSON de envío de datos.

Lo envía con el siguiente formato:

```
{  
  "state": " Estado final (win / lose)  ",  
  "nivel": " Nombre de la escena de la sesión ",  
  "usuario": " Usuario ",  
  "devClass": "Codigo de aplicación ",  
  "time_resolve": " Tiempo de resolución ",  
  "objetos": [] " nombre de objetos, movimientos del objeto ",  
  "movimientos_tot": " Numero de movimientos totales de los objetos "  
}
```

6.2. Creación de Juego y Experimento en el sitio web

Lo primero que tenemos que hacer para poder crear o añadir un juego a la base de datos es registrarnos en nuestra web como Investigador. Para ello accedemos a

```
{http://monitorizer.sytes.net/main.html#/}
```

donde lo primero que vemos, si no tenemos datos de una sesión anterior en la caché, es lo siguiente.

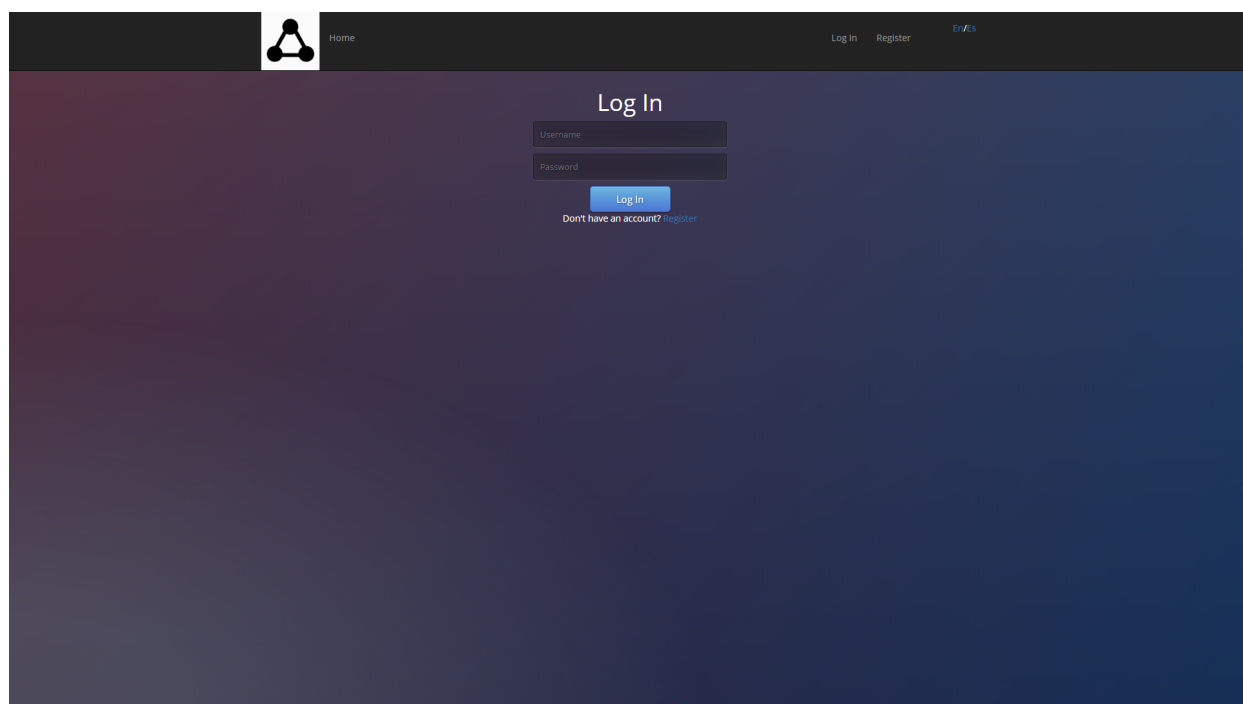
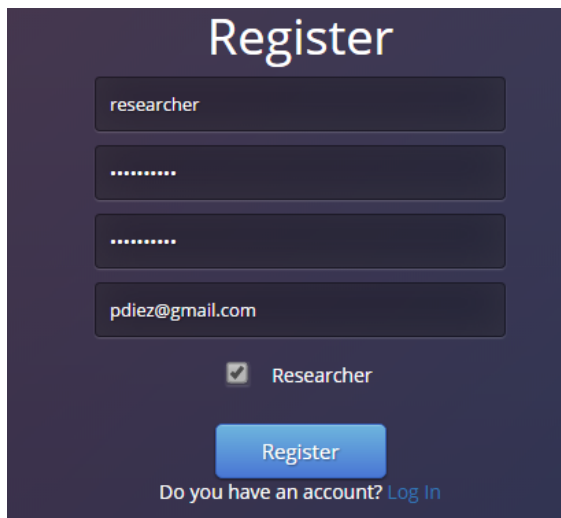
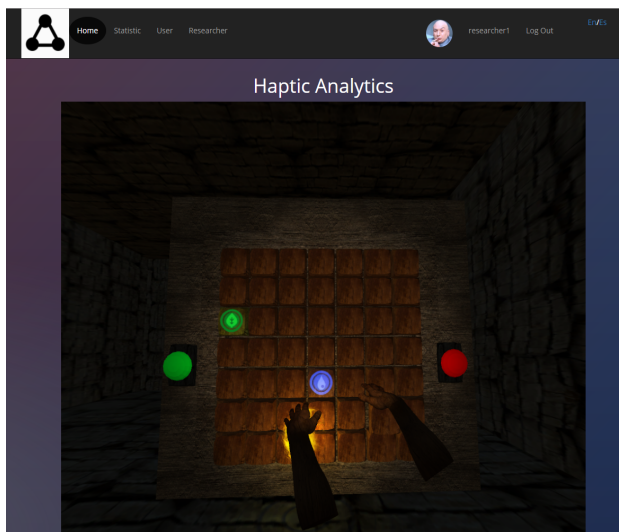


Figura 6.3: Pantalla principal.

A continuación vamos a la parte de registro donde aparece un formulario a rellenar 6.4a. Una vez completado los campos, sin olvidarnos de pulsar el *checkbox* de Investigador, pulsamos sobre registrar y si todo ha ido correctamente, debería aparecer algo similar a la Figura 6.4b.

A dark-themed registration form titled "Register". It contains four input fields: a username field with "researcher", two password fields with masked characters "*****", and an email field with "pdiez@gmail.com". Below the email field is a checkbox labeled "Researcher" which is checked. At the bottom is a blue "Register" button and a link "Do you have an account? Log In".

(a) Formulario de registro.



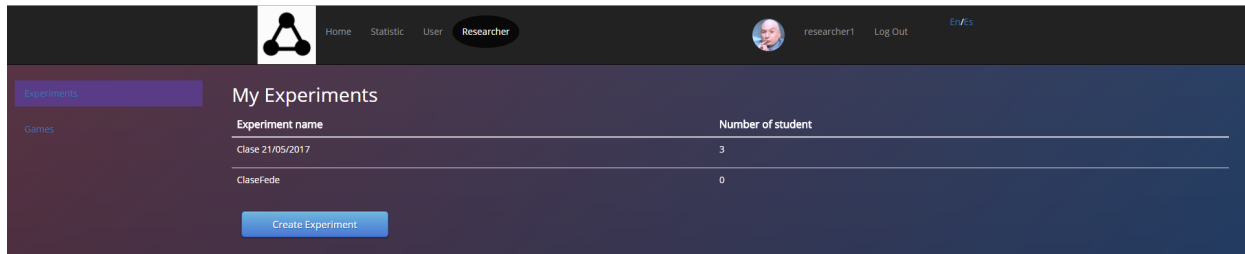
(b) Pantalla una vez registrado.

Figura 6.4: Pantalla de registro y una vez registrado.

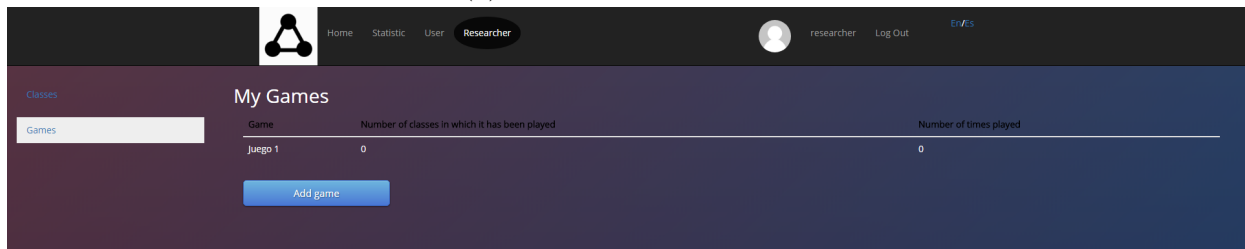
Desde aquí podemos acceder al menú de Investigador, el cual contiene a la izquierda un submenú con los dos apartados clave:

- Experimento
- Juegos

Ambos siguen la misma estructura, que consiste en una tabla con los Experimentos o Juegos creados, en caso de no que se hayan creado alguno se mostrará un mensaje, y un botón para añadir. Si pulsamos sobre el botón *Crear Experimento* nos aparecerá un formulario con un solo campo para introducir el nombre que tendrá. Con el Juego es semejante. Una vez rellenamos ambos nos debe aparecer algo parecido a las siguientes figuras.



(a) Experimentos creadas.



(b) Juegos creados.

Figura 6.5: Pantallas de Experimentos y Juegos creados.

Para acceder a los detalles de un Juego o un Experimento simplemente hace falta hacer clic sobre la fila, en nuestro caso iremos a los detalles de Clase 1, donde nos aparece toda la información relacionada con ella y lo que es más importante de todo, su código. Éste código es el que entregamos a los usuarios del que obtendremos los datos y que tienen que introducir una vez arranquen el juego. Esta página es similar a la siguiente imagen.

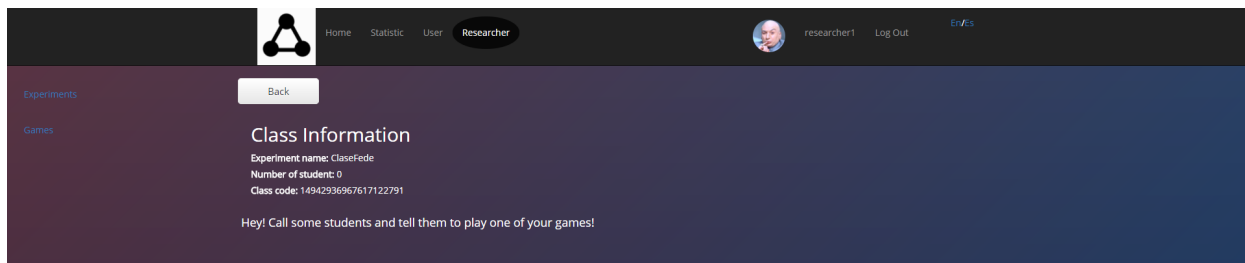


Figura 6.6: Detalles de Clase 1.

6.3. Las fases del juego

Las principales fases por las que pasa el jugador de *Escape Room* son dos, la parte de pulsar los botones de la sala 1 en el orden correcto para escribir la palabra ABRIR y en la segunda sala el pulsar los botones del piano en el orden correcto.



Figura 6.7: detalle de la segunda sala.

6.4. Visualización de los resultados

Una vez se ha mandado al servidor la petición de añadir la Sesión, tanto el jugador como el investigador podrán ver los datos de ésta.

En el caso del jugador tiene que acceder al menú de Estadísticas en el que tiene dos submenús:

- Ránking
- Gráfica

En este caso accedemos al apartado de Gráfica, en el que podemos ver los objetos usados y el tiempo que ha tardado desde que arrancó la sesión hasta que se envía la petición. En nuestro juego, la petición se envía cuando el usuario termina las pruebas, por lo que el dato realmente es el tiempo que tarda el jugador en pasarse el juego.

Para poder ver las gráficas el usuario debe escoger un Juego de los que ha jugado. Una vez escogido, se le da la opción de escoger una Sesión concreta para poder ver de esta manera los objetos que ha usado únicamente en esa Sesión. En caso de no escoger ninguna, se mostrarán la suma de todas las sesiones en la gráfica de objetos usados.

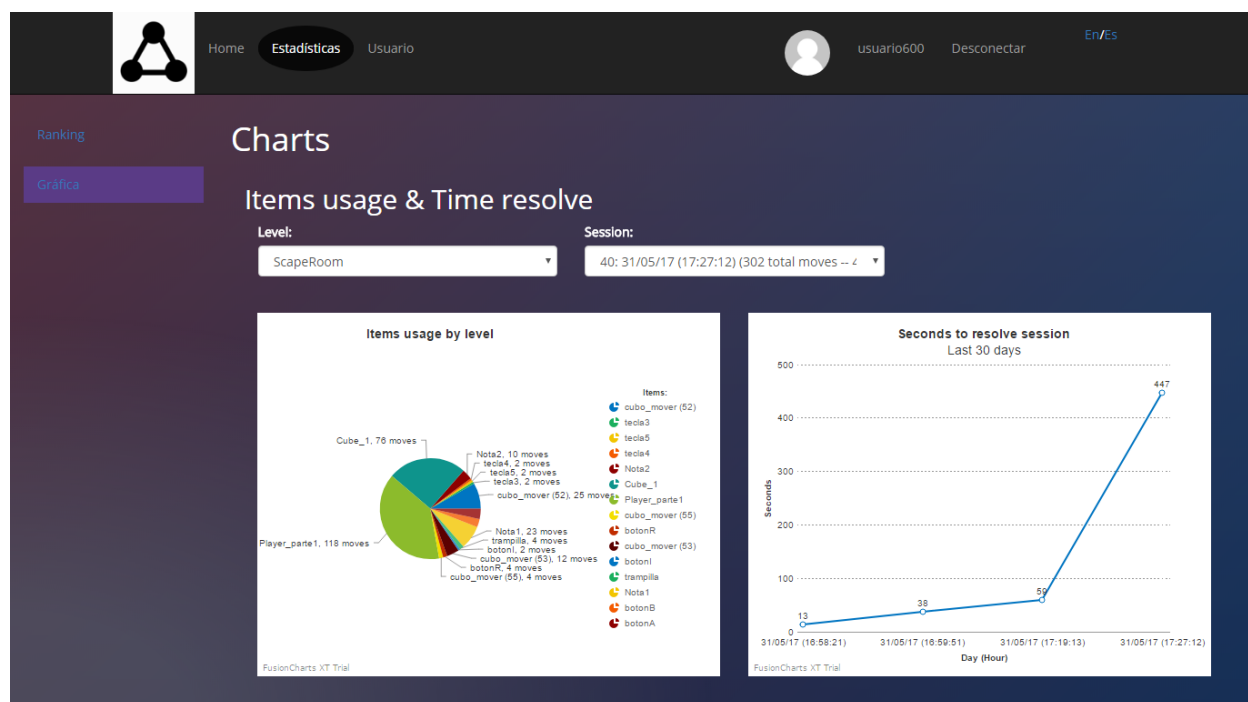


Figura 6.8: Gráfica de una Sesión concreta del Juego Escape Room.

Por otro lado, si somos un Investigador podremos acceder bien a los experimento que hayamos creado o bien a los Juegos que hayamos añadido.

Dentro de los detalles de una Experimento tenemos acceso, además de los datos propios de ese Experimento, a una tabla en la que se muestran las sesiones en las que se ha jugado a cualquier Juego pero dentro del mismo Experimento. En esta misma pantalla también podremos ver una gráfica del número de veces que se ha jugado a cada Juego en ese Experimento.

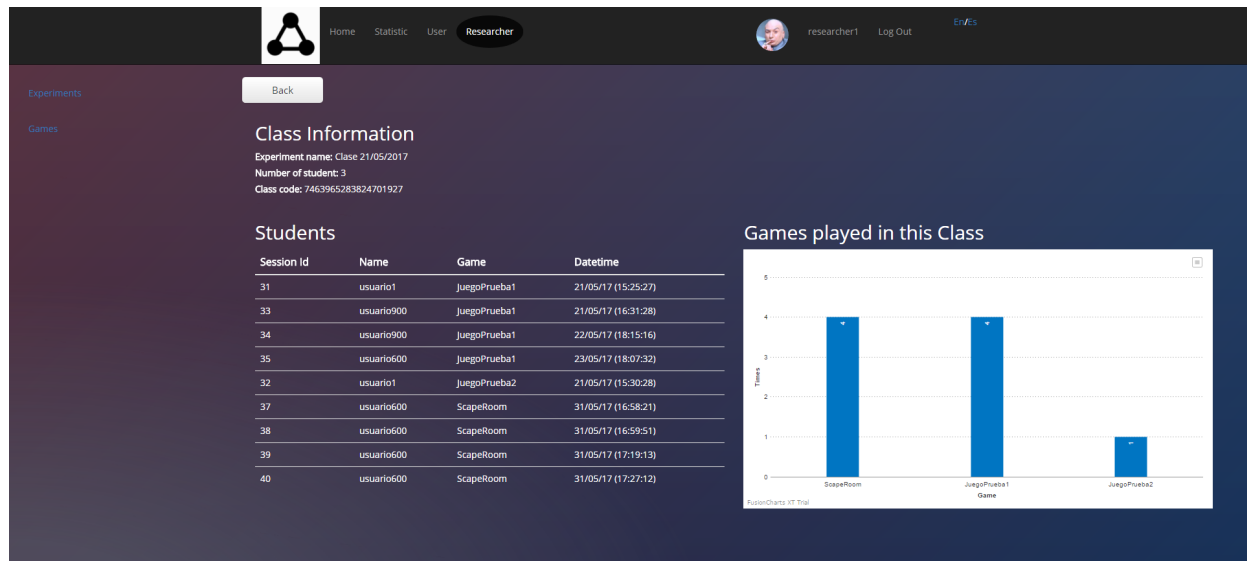


Figura 6.9: Detalles de un Experimento con datos.

En el caso de que accedamos a los detalles de un Juego veremos los Experimentos en las que se haya jugado, haya sido o no creada por nosotros, y de nuevo dos gráficas iguales a las mencionadas en el caso de acceder al submenú Gráfica. La diferencia que existe respecto a esas son los filtros a los que están sometidos para conseguir los datos. En esta ocasión se podrá filtrar por Experimento, por Usuario o por Experimento y Usuario

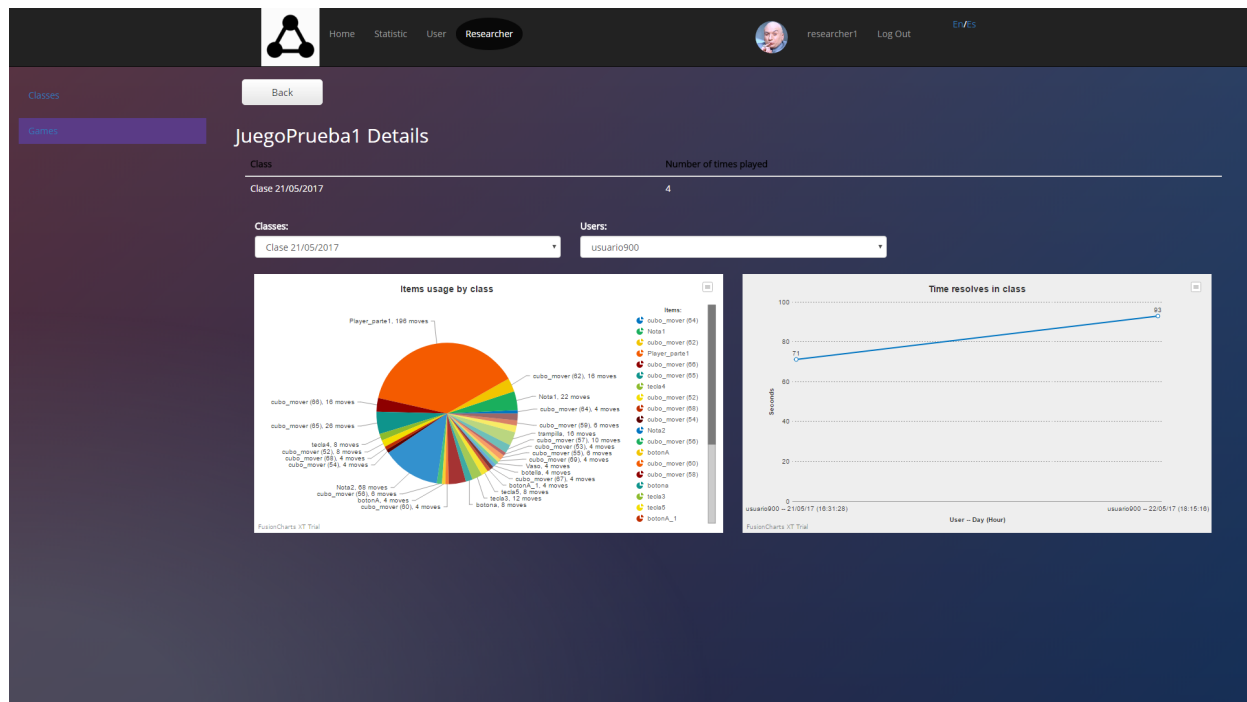


Figura 6.10: Detalles de un Juego filtrado por Experimento y por Usuario.

Las gráficas de la sección Investigador cuenta con la opción de descarga en forma de imagen (PNG y JPG), documento PDF, gráfico vectorial (SVG) ú hojas de cálculo (XLS). Ésta opción está situado en la esquina superior derecha de cada gráfica. La sección dedicado a las estadísticas individuales no cuenta con esta opción.

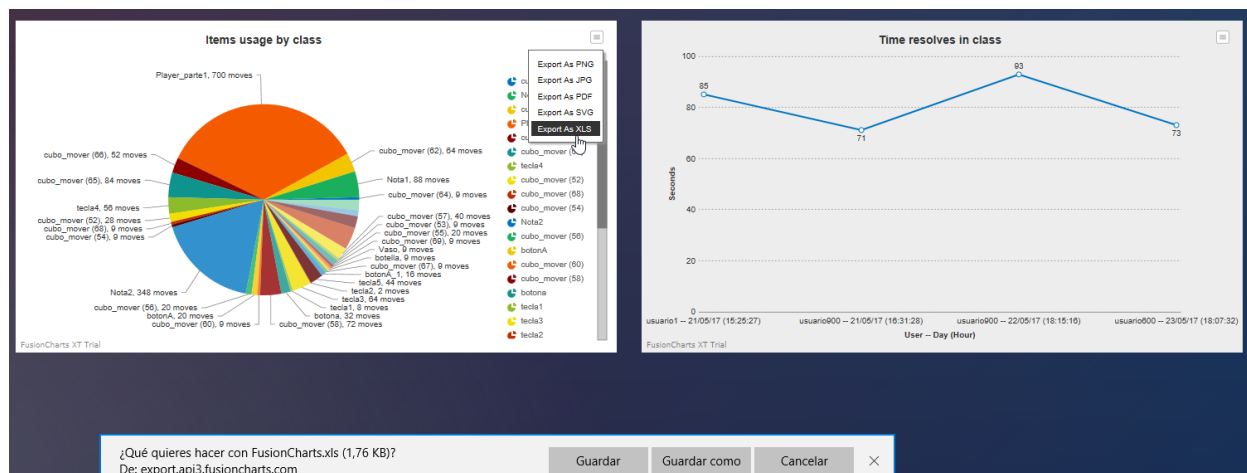


Figura 6.11: Descarga de la gráfica en diferentes formatos.

Capítulo 7

Conclusiones

A lo largo de este proyecto hemos trabajado para dar soporte a la investigación del comportamiento de los usuarios en aplicaciones de RV. *Haptic Analytics*, el sistema desarrollado, es una herramienta de recolección y análisis de datos de la experiencia del usuario en aplicaciones que cuentan con seguimiento de movimiento del visor y seguimiento de posición y reconocimiento gestual de las manos.

En la presente memoria hemos expuesto cómo estos investigadores pueden crear fácilmente sus experimentos desde el sitio web de *Haptic Analytics*, obteniendo unos códigos que permiten identificar la aplicación de RV que va a ser objeto de estudio, y el grupo de usuarios que van a participar en un determinado experimento. Cuando los usuarios utilicen dicha aplicación, esta llevará integrado un componente adicional con un API que permitirá agrupar y enviar en tiempo real los datos relativos a la experiencia de usuario al servidor de *Haptic Analytics* a través de la Web. Los datos enviados por el cliente, son almacenados en el servidor para su posterior consulta, visualización, y descarga en forma de hojas de cálculo o imágenes preparadas para su publicación. Los datos son accesibles tanto a los investigadores, que pueden agregarlos por aplicación, por experimento o por sesiones individuales de uso, como a los propios usuarios que pueden estar interesados en conocer sus propias estadísticas y algún dato agregados del total de usuarios que han participado en su mismo experimento.

La implementación actual consiste en un complemento de Unity, que debe ser integrado en la aplicación que hará de cliente, y un servidor ubicado en una de nuestras máquinas donde se aloja un sitio web. La parte delantera utiliza AngularJS, Bootstrap para los estilos y FusionChart para las gráficas, y cuya parte trasera ha sido desarrollada sobre Django, con un API construida con Django Rest Framework para realizar las llamadas de añadir y consultar sobre las entidades, y conectada con una base de datos PostgreSQL.

Las pruebas realizadas sobre las dos aplicaciones desarrolladas, *Cube Board* y *Escape Room*, ambas orientadas a Oculus Rift y Leap Motion, confirman que es un sistema plenamente funcional, que permite recolectar y enviar al sitio web toda la información que posteriormente el investigador podrá consultar y visualizar de manera sencilla y práctica.

Los resultados, que resumiremos a continuación, apuntan a que esta puede ser una herramienta muy útil para agilizar futuras investigaciones en el campo de la Realidad Virtual.

7.1. Resumen de resultados

Estos son los objetivos específicos que hemos alcanzado:

- Desarrollo de un sistema informático que sirve como herramienta de recolección y análisis de datos de la experiencia de usuario.
- Creación de una primera aplicación de ejemplo para probar el sistema, el puzzle *Cube Board*.
- Creación de una segunda aplicación de ejemplo para probar el sistema, el videojuego *Escape Room*.
- Desarrollo de la parte trasera del sitio web que gestiona la información guardada de los experimentos y los usuarios.
- Desarrollo de la parte delantera del sitio web donde usuarios e los investigadores acceden y visualizan los datos del sistema.

Como aspectos positivos del desarrollo del proyecto, además de haber experimentado en primera persona todas las fases de este amplio proceso, podemos señalar los siguientes.

Por un lado, hemos podido aprender y profundizar en el uso de las tecnologías empleadas en las diferentes capas del sitio web, como AngularJS, JQuery, y Django, estructurando el código según diferentes niveles de abstracción. Para trabajar tanto en la parte frontal como en la trasera del sitio, hemos optado aprender y usar PyCharm, que tiene herramientas de control de versiones y depuración. Y ya que necesitábamos acceder al servidor desde cualquier lugar, hemos aprendido a usar las DNS para tener un acceso fijo a través de un enlace de un servicio externo gratuito.

Por otro lado, hemos profundizado en el desarrollo de videojuegos y en la programación de *scripts* en Unity, abordando cuestiones particulares de la RV. Concretamente hemos trabajado con el casco Oculus Rift y el reconocedor de manos Leap Motion, que eran tecnologías sobre las que existía mucho interés por parte de los autores del trabajo.

Los aspectos negativos del desarrollo del proyecto se señalan a continuación.

Uno de los mayores problemas que tuvimos fueron los cambios en los requisitos, que aparecieron de improviso hasta casi la mitad del proyecto. Fue necesario eliminar aquellas funcionalidades que no eran compatibles con los nuevos requisitos, aunque la mayoría se pudo reutilizar o adaptar para crear una nueva versión.

Al no poder contar con un ejemplar de Oculus Rift y de Leap Motion desde el principio, tuvimos que hacer numerosos cambios en las aplicaciones desarrolladas con Unity llegado el momento de la integración. Lo más problemático con diferencia fue Leap Motion ya que a menudo no era capaz de detectar posiciones o gestos de las manos debido a diversos factores del entorno, como la condición lumínica, y no fue posible diseñar una manera más cómoda para que el jugador se moviese, quedando vigente el método de cerrar el puño izquierdo, dejando libre la mano derecha para poder interactuar con objetos.

También tuvimos problemas con la implementación del servidor web, ya que la mayoría de los autores no conocíamos el lenguaje de programación Python y mucho menos la tecnología de Django. El servidor se volvía inestable y dejaba de funcionar pasado cierto tiempo si lo dejábamos lanzado desde el entorno de desarrollo de PyCharm, siendo la solución ejecutar el servidor desde una consola de comandos.

7.2. Trabajo futuro

Finalmente vamos a explicar aquí las líneas de trabajo futuro que consideramos más prometedoras, tanto las que implican mejorar el cliente y el servidor de *Haptic Analytics* como las que profundizan en los dos juegos desarrollados como aplicaciones de referencia del sistema.

Con respecto al componente adicional desarrollado en Unity para la parte del cliente, sería interesante facilitar al máximo su instalación y uso por parte de desarrolladores que no conocen nuestro sistema, documentando mejor el software e incorporando alguna manera más sencilla de llamar al API de *Haptic Analytics*. También sería necesario identificar mejor cada sesión individual que el usuario pase experimentando con una aplicación.

Como mejoras de la primera aplicación de ejemplo, *Cube Board*, se podría simplificar la lógica del juego para hacerla más comprensible e incluir gestos para coger y soltar piezas del tablero o la torre. Como mejoras de la segunda aplicación de ejemplo, *Escape Room*, se podrían añadir nuevas estancias al escenario, o incluso generar procedimentalmente nuevas versiones de las mismas. La mejora más obvia consistiría en ampliar el trabajo con una investigación sobre una aplicación real y tomando alguna medida objetiva como el tiempo promedio de ejecución de una tarea.

Con respecto a la parte del servidor, tanto en la capa de presentación como la capa de acceso a datos, se nos ocurren muchas vías interesantes a explorar. Se podrían añadir más filtros en las gráficas para obtener datos más concretos y personalizados para las aplicaciones de los investigadores (filtro por sexo, rango de edad, etc.). Se podrían también añadir datos a las propias aplicaciones como por ejemplo el género de un videojuego o las plataformas en las que se puede ejecutar la aplicación.

Esto implica otro cambio en las sesiones de uso, concretamente en el modelo *Session Details*, donde se podría incluir en que plataforma se ha usado la aplicación y así después poder filtrar por ello. En cuanto al modelo interno del servidor, se podría crear una nueva entidad, por ejemplo denominada Proyecto, la cual estaría relacionada con la entidad Experimentos en una relación de 1 a N.

Hay muchas mejoras posibles en la generación de tablas y gráficas. Como por ejemplo añadir un apartado nuevo en el perfil del Investigador que fuera un *dashboard*, en el que poder crear y gestionar gráficas propias a través de combinar varios elementos, como por ejemplo el tipo de gráfica (*Column and Bar*, *Line and Area*, *Pie Chart Multi-Level Pie*). Otra mejora es la inclusión de un botón de Favorito en las gráficas y tablas para que aparezcan automáticamente en el *dashboard*. Esto ayudaría al investigador a consultar los datos más importantes, según su criterio, de una manera más rápida, sin necesidad de encontrar la gráfica o la tabla adecuada deambulando por las distintas secciones fijas del sitio web.

Todo ello supone crear dos nuevas entidades en la parte trasera del sitio web, como por ejemplo Dashboard, que conecta investigadores con las gráficas, y Chart, que guarda la configuración propia (nombre, tipo, etc.).

Aportaciones individuales de los autores

Pablo Diez de la Puente

Mis aportaciones en general han ido orientadas a la parte trasera (*back-end*) y a la visualización de los datos en la parte delantera (*front-end*) del sitio web. Además me he encargado tanto de preparar el repositorio en GitHub, para que pudiéramos acceder todos los componentes del grupo al código referente con la aplicación web, como de buscar la plantilla de la memoria e implementarla en Overleaf.

Más concretamente, durante la etapa de investigación, me encargué de hacer la especificación de las variables que podríamos llegar a obtener una vez se conocía la temática y funcionalidades del juego *Cube Board*. A su vez, investigué acerca de la instalación del *framework* Django y el montaje del servidor en mi ordenador. Esto incluye desde relacionar las bases de datos con el framework hasta desplegar el proyecto escogiendo el dominio de DDNS, las DNS para el servidor y el redireccionamiento necesario de puertos en mi *router*. Una vez el servidor web Angular JS estaba implementado junto a la autenticación de usuarios, empecé a buscar distintos *plugins* de gráficas para la muestra de datos.

Durante la parte de desarrollo, me encargué de las tablas y relaciones entre ellas una vez definimos las entidades que iban a existir en nuestra aplicación. Acto seguido comencé a implementar las llamadas al *back-end*, para que se pudieran usar tanto desde los juegos como desde el sitio web separando de esta manera las llamadas al servidor entre las de adición de contenido y entre las de obtención de datos.

En relación con el *front-end*, añadí el apartado para el investigador donde se empezarían a utilizar más llamadas al servidor. En paralelo con esta nueva parte comencé a desarrollar las gráficas que contienen la sección de estadísticas y la sección del investigador. Estas gráficas cambian la información dependiendo de distintos filtros que les acompañan, los cuales también requieren información dinámica que varían según el contenido registrado previamente por el investigador.

En lo referente a la memoria, mi aportación ha sido continua a lo largo del año, empezando el primer tercio del curso haciendo el capítulo 1, el capítulo 2 y el capítulo 3 junto a mis compañeros. Durante el segundo trimestre me dediqué a crear las citas y referencias que hacíamos en esos tres primeros capítulos con el mismo estilo además de añadir contenido para los dos siguientes capítulos. Lo que restaba de curso lo dediqué a escribir parte de los últimos capítulos, concretamente el capítulo 6 y el capítulo 7, junto a las demás secciones de la memoria como por ejemplo los anexos, el resumen y mis propias aportaciones.

David Valdivia Martinez

Mis aportaciones en general han estado relacionadas principalmente con la aplicación desarrollada en Unity, la investigación y el desarrollo de aplicaciones relacionadas con el uso de dispositivos de RV y Leap Motion, así como el análisis del desarrollo del modulo de recogida y envío de información desde Unity.

En la etapa de investigación del juego *Cube Board* concrete la idea principal del juego, y la posible información que podría enviarse desde él, también insistí en el uso de GitHub, herramienta que finalmente utilizamos para las distintas partes del proyecto.

En la etapa de desarrollo, primero me ocupé de hacer el juego sencillo llamado *Cube Board*, para más tarde comenzar con el modulo de envío de información y seguidamente aplicarlo a un juego mas complejo, como es *Escape Room*, finalmente terminé verificando que todo funcionaba correctamente en un juego ajeno a nosotros descargando de la Asset Store de Unity, un template llamado TopDown_AI.

Modele y texturice todos los componentes de el primer juego, ademas de desarrollar toda la lógica del juego. Al no tener en un principio el dispositivo Oculus Rift realice varias versiones con el uso solamente de Leap Motion situado enfrente del jugador. Cuando pude hacerme con el dispositivo, inicie la etapa del desarrollo propiamente dicha en RV, integrando todo lo hecho hasta el momento.

Mejore y adapte a RV tanto el *script* de texto, para poder realizar tutoriales, como la primera escena de inicio de sesión desde la aplicación.

En la parte de desarrollo del segundo juego, *Escape Room*, no pude contar con el suficiente tiempo como para poder modelar todos los componentes del juego en Blender, por lo que tuve que buscar y adaptar modelos encontrados en la Asset Store de Unity. Me ocupe de toda la construcción del juego y la posterior adaptación a RV.

Realice el modulo de recogida de información que buscaba todos los objetos interactivos en la escena, y les asignaba automáticamente un *script* que analizaba si se estaban moviendo y asignaba una variable numérica a ese movimiento para que cuando terminara la sesión enviarlo al servido, posteriormente lo mejore incorporando un filtro por nombre completo o contenido en el nombre de los objetos de la escena, para que fuera mucho mas sencillo filtrar los objetos mas importantes de la escena. Finalmente añadí funcionalidades para añadir variable y movimiento ajenos al funcionamiento principal del módulo.

En el trabajo realizado en la memoria, desarrolle conjuntamente la mayoría de los aparatos, sobretodo los que influían directamente en la aplicación desarrollada en Unity.

Aparte asistí a la mayoría de las reuniones y videoconferencias organizadas por el grupo de trabajo, y también he participado activamente en las decisiones y opiniones del grupo.

Sheng Zhu

En el primer cuatrimestre, nos enfocamos más en la parte del videojuego y del módulo. Mi trabajo consistía en realizar un *script* (al que denominé iterador de texto) que permitía disponer de un conjunto de texto que se muestra en la pantalla y que al cabo de un tiempo o apretando una combinación de teclas, permitía avanzar al siguiente texto o si cumplía ciertas condiciones, avanzar a otra escena.

En el siguiente cuatrimestre, después de presentar nuestro estado del proyecto en la primera reunión, empezamos a dividir el trabajo, ya que abordamos más contenido a parte del videojuego y del módulo. Mi parte del trabajo consistía principalmente en el diseño y el desarrollo de la aplicación web que se encarga de mostrar los datos de forma simple y organizada.

Tras repartir las tareas, empecé una nueva etapa de investigación para determinar las tecnologías que usaría en aplicación web. El resultado de esta investigación resultó en el uso de AngularJS como tecnología principal de la aplicación, y JQuery para facilitar el manejo de los elementos DOM.

Al comienzo de la etapa, he realizado un diseño del sitio web que mis compañeros han revisado y sobre el que han dado el visto bueno, por lo que proseguí a maquetar el frontal de dicha web.

Tras la segunda reunión, implemente el sistema de traducciones en tiempo real en la web, poniendo a disposición del usuario dos idiomas, el inglés y el español.

Después de haber implementado el sistema de traducciones, empecé a implementar el sistema de autenticación en el *front-end* usando la tecnología de JWT con la ayuda de la librería de Satellizer. Por último, implementé el sistema de notificaciones, para que muestre una pequeña descripción en caso de algún error. Una vez implementado el sistema de notificaciones, proseguí con el *back-end*.

En el *back-end*, adapté el modelo de usuario por defecto que posee Django a lo necesario para nuestro proyecto, desarrollé las funciones necesarias para la API para tratar los datos de los usuarios e implementé un nuevo modelo para contener los datos del experimento y así poder gestionar los experimentos de los investigadores.

Una vez implementado y probado las nuevas funcionalidades, decidí hacer un cambio en los estilos de la aplicación web. Uno de los cambios destacables es la implementación del *framework* Bootstrap para los distintos elementos preconfigurados como lo son los botones o los campos de entrada.

Tras realizar ajustes en los estilos, me dispuse a implementar los dos roles que existe ahora en la aplicación, el jugador y el investigador, teniendo, el segundo, privilegios para crear experimentos.

Al tener la aplicación funcional, me dispuse a realizar el diseño de la pantalla de inicio en el videojuego donde es una pantalla de inicio de sesión, una pantalla de registro y otra pantalla para introducir el código de experimento, así como toda la lógica de estas pantallas para conectar con la base de datos.

En paralelo a toda la etapa de desarrollo, he estado, junto con mis compañeros, escribiendo la memoria del proyecto. En el primer cuatrimestre empezamos a elaborar los 3 primeros capítulos que está compuesto por la introducción, estado del arte y los objetivos.

En el segundo cuatrimestre, he elaborado conjuntamente los capítulos 5, 6 y 7.

Después de tener un proyecto entregable, me dispuse a revisar todo el documento y mejorar y corregir los fallos.

Aparte de todo esto, he participado en la traducción del Resumen, y de los capítulos de Introducción y Conclusiones.

He asistido a todas las reuniones de seguimiento propuestas por el director y el codirector, aparte de las reuniones organizadas exclusivamente entre los miembros del proyecto. También he participado en la toma de decisiones respecto al curso del proyecto.

Referencias

- BeautiFun Games. (2012). *Nihilumbra*. [Multiplataforma].
- Blender Foundation. (1995). *Blender*. Descargado de <https://www.blender.org/>
- Blizzard Entertainment. (2014). *Hearthstone: Heroes of warcraft*. [Microsoft Windows, macOS, iOS, Android].
- Cooper, E. (2016). *Eve valkyrie*. [PlayStation 4, Microsoft Windows].
- Crytek. (2016). *The climb*. [Microsoft Windows].
- Digital Confectioners. (2017). *Depth vr*. [Microsoft Windows].
- EA Canada, EA Bucharest. (2016). *Fifa 17*. [Multiplataforma].
- EA Digital Illusions CE. (2002). *Saga battlefield*. [Multiplataforma].
- Eligogames. (2016). *Hellvibe*. [Microsoft Windows].
- Furness, T. A. (1981). *Cabina virtual*.
- GameTop. (2011). *Pharaoh puzzle*. [Microsoft Windows].
- Hoffman, H. (2015). *Aboard the looking glass*. [Microsoft Windows].
- Namco Bandai Games. (2015). *Tekken 7*. [Multiplataforma].
- Niantic Inc. (2016). *Pokemon go*. [Android, iOS].
- Obsidian Entertainment. (2015). *Pillars of eternity*. [Microsoft Windows, OS X, Linux].
- Rocksteady Studios. (2016). *Batman arkham vr*. [Playstation 4, Microsoft Windows].
- Schubert, M. (2014). *Weightless*. [Microsoft Windows].
- SuperHOT team. (2016). *Superhot vr*. [Microsoft Windows, MacOS, Linux].
- Sutherland, I. E. (1965). The ultimate display. En *Proceedings of the ifip congress* (pp. 506–508).
- Toxic Games. (2016). *Q.u.b.e*. [Multiplataforma].
- Wood, B. (2016). *2016 is poised to be the year that consumer awareness of virtual reality reaches meaningful levels*.

Apéndice A

Anexo 1. Introduction

The Virtual Reality (VR) boom that we have nowadays was originated in 2009 when Palmer Lucky, a passionate collector, tried to shape his idea of creating the cheapest and most advanced VR helmet. His project, Oculus Rift¹ got an overwhelming funding thanks to the collective funding platform Kickstarter². He attracted veterans from the industry, like John D. Carmack, who got involved in the new company, and Facebook bought it for 2000 million dollars.

Both the media and the investigation director from CSS Insight declared 2016 as “the year in which the consumer’s conscience on Virtual Reality reaches significant levels”, (Wood, 2016) because it coincided with the launching into the market of a whole new generation of VR helmets, like Oculus Rift itself, HTC Vive³ and Playstation VR⁴. Eventhough it is not an innovative idea, as we will see when we study the phenomenon, in some ways these last years have meant a revival of the economic investment and the public’s interest on the devices and applications related to VR: Facebook with Oculus Rift, Valve and HTC with HTC Vive and Sony with its own PlayStation VR.

¹Oculus Rift: <https://www.oculus.com/rift/>

²Kickstarter: <https://www.kickstarter.com/>

³Vive: <https://www.vive.com/eu/>

⁴PlayStation VR: <https://www.playstation.com/es-es/explore/playstation-vr/>

These three devices, that could be classified as high-end, are the ones that have had the most media influence. However, cheaper options with different performance can be found in the market, from the simple mobile phone viewers, like GearVR⁵, Cardboard⁶ or Daydream⁷ from Google, and other mobile phone viewers, of which there are infinite options, to the showy physical yincanas like The Void⁸, with groups of players united in their installations wearing viewers and haptic controllers, without forgetting about the most popular images and spherical, or immersive, videos taken using an omnidirectional view of 360 degrees.

The development studios and digital content editing companies look to apply new Virtual Reality experiences in fields like Entertainment, Advertising and Communication. This technology has specifically posed the challenge of reaching the users mainly through videogames, with titles such as *The Climb* (Crytek, 2016), *EVE Valkyrie* (Cooper, 2016) or *Superhot VR* (SuperHOT team, 2016) Where precision is not a requirement and where health or business are not at risk, unlike in other applications of which the purpose is away from fun and leisure.

In order to innovate and to take advantage of the technology, both the creators and the investigators need to know in depth the way in which users interact with these applications, the way they enjoy and benefit from them. Without a doubt, having access to objective and exhaustive information about this behaviour allows an improvement on the software, strengthening or restricting different aspects of the experience depending on how the significance for the real users is valued. For example, knowing which levels, characters or objects from videogames were liked more or less by players in order to create more interesting, striking or even adapted titles.

As a premise to this project we considered that the more knowledge is acquired during the tests (carried out in each iteration in case of iterative development processes) the quality from the final result and the obtained software can be more satisfactory.

⁵Gear VR: www.samsung.com/GearVR

⁶Google Cardboard: <https://vr.google.com/cardboard/>

⁷Google Daydream: <https://vr.google.com/daydream/>

⁸The Void: <https://www.thevoid.com/>

The motivation for this project arose when we became aware of this need, which is not just a demand from the industry, but it also seems apparent for an investigating team like the one in which our director is currently working, where they look into narrative videogames and new technologies like VR. Taking all of this into account, the tool we have set out to develop is a computer system that will give support to the investigation on the user's real behaviour towards advanced VR applications, specifically games or applications that carry out both the monitoring of the viewer movement and the tracking of the position and recognising hand gestures.

Therefore, Haptic Analytics will be responsible for, in the first place, harvesting information on the user (how he manages through the game, where does he draw his attention or with which objects does he interact and how). In order to succeed in this first information collection, we will create at least one software component specific to that system that can be integrated into the application itself under study, as well as a few example applications that have it implemented already and make use of it, to have them as a reference.

In second place, the system will offer investigators, once the experiments have been performed on real users, a website where they can access easily and at any time, with authentication, all that information they wish to consult, visualise or download, in the form of tables or graphs with added data depending on the criteria, for future process and analysis.

After explaining the context of this Project in this chapter, we will proceed to present the structure of the report. Chapter 2 examines the state of the techniques in VR, the devices, development environments and existing tools, as well as some videogames that we have used as a reference for the development of our example applications. In chapter 3 the objectives for this project are detailed, making specifications of the requirements that will be used as a basis for further development. The general work plan, the tools related to communication and the work stream used in the project are presented in Chapter 4, as well as the small influences from different methodologies. Chapter 5 contains the majority of the contribution of this project, as it is where the analysis, design and some details about the implementation of the proposed system, *Haptic Analytics*, are detailed. Everything related to the development of both model applications, which show the use of this system, is also included in this chapter. The tests performed over these application and the results obtained from them are discussed in Chapter 6. All the different conclusions on the objectives achieved during the project and the aspects that need to be improved in the future are explained in Chapter 7. Finally, we put together our conclusions on this project, that has had positive results and has pointed out several interesting lines of work to continue improving the tool in the future, and the various annexes.

Apéndice B

Anexo 2. Conclusions

Throughout this project we have worked to support the investigation of user's behaviour in VR applications. Haptic Analytics, the system developed, is a tool for data collection and user experience's analysis in applications that have movement and position tracking of the devices and hand gesture recognition.

In this paper we have described how these researchers can easily create their experiments from the Haptic Analytics website. obtaining some codes that allow to identify the VR application that will be object of study and the group of users that are going to participate in a given experiment. When users use this application, it will be integrated an additional component with an API which will allow to group and send data related to the user experience to the Haptic Analytics server. The data sent by the client are stored on this server for later consultation, viewing and downloading as spreadsheets or images prepared for publication. These data are accessible both by researchers, who can filter them by application, by experiment or by individual sessions of use, such as the users themselves who may be interested in knowing their own statistics and some aggregated data of the total users who have participated in the same experiment.

The current implementation consists of a Unity plugin-in, which must be integrated in the application as the client part, and a server located in one of our computers where the website is hosted. The front-end uses Angular JS, with Bootstrap for styles and FusionChart for graphics, and the back-end has been developed in Django with an API built with Django Rest Framework to make add and query requests on entities. This back-end is connected with a PostgreSQL database.

Tests conducted on the two applications developed, Cube Board and Escape Room, both oriented to Oculus Rift and Leap Motion, confirm that Haptic Analytics is a fully functional system that allows the client collect and send all the information. The researcher can consult and display the data later in a simple and practical way.

The results, which are summarized below, suggest that Haptic Analytics can be a very useful tool to streamline future research in the field of virtual reality.

B.1. Summary of result

These are the specific objectives we have achieved.

- Development of a computer system that serves as a tool for collecting and analyzing user experience data.
- Creating a first sample application to test the system, the Cube Board puzzle.
- Creating a second sample application to test the system, the Escape room.
- Development of the website's back-end that manages the stored data of the experiments and the users.
- Development of the website's front-end where users and researchers access and visualize the system data.

As positive aspects of the development of the project, in addition to having experienced in first person all the phases of this broad process, we can mark the following ones.

On the one hand, we have been able to learn and deepen the use of technologies used in front-end and back-end of the website, such as AngularJS, JQuery and Django, structuring the code according to different levels of abstraction. To work on both sides of the website, we have chosen Pycharm, which has version control and debugging tools. Since we needed to access the server from anywhere, we learned to implement the DDNS to have a domain and reach to it through a link.

On the other hand, we have deepened in the development of video games and programming Unity scripts, addressing particular issues of VR. Specifically, we have worked with Oculus Rift and the hand recognition device Leap Motion, which were technologies that had a lot of interest on the project components.

The negative aspects of project development are outlined below.

One of the biggest problem we had were the changes in requirements, which suddenly appeared in almost half of the project. It was necessary to eliminate those functionalities that were not compatible with the new requirements, although the majority could be reuse or adapted to create a new version.

Since we could not rely on a kit of Oculus Rift and Leap Motion from the beginning, we had to make numerous changes to the applications developed with Unity at the integration time. Leap Motion was most problematic because it was often unable to detect hand positions or gestures due to various environment factors, such as light conditions, and it was not possible to design a more comfortable way for the player to move than closing the left fist, leaving the right hand free to interact with objects.

We also had problems with the web server implementation, since most of the authors did not know Python and much less Django technology. The server became unstable and stopped working after a certain time if we launched it from PyCharm IDE instead of run the server from a command console.

B.2. Future work

Finally we are going to explain here the future lines of work that we consider more promising, both those that imply improving the client and the server sides of Haptic Analytics as those that deepen in the two applications developed.

In reference to the additional component developed in Unity for the client side, it would be interesting to facilitate its installation and use by developers who do not know our system. documenting the software in a better way, including the Haptic Analytics API. It would also be necessary to better identify each individual user session.

As improvements of Cube Board, we could simplify the game logic to make it more understandable and include catching gestures or dropping pieces from the board and tower. As improvements of Escape Room, new rooms could be added to the stage, or even new versions of the same. However the most obvious improvement would be to extend the work with research on a real application and to take some objective measure such as the average time of execution of a task.

With regard to the server side, in both the front-end and the back-end, we come up with many interesting paths to explore. More filters could be added to the graphs to obtain more specific and personalized data for researchers applications (gender filter, age range, etc). We could also add data to the applications themselves such as the genre of a video game or the platforms on which the application can run.

This changes implies another change in the use sessions, specifically in the model Session Details, where you could include in which platform the application has been used and thus to filter for it. In reference to the server internal model, a new entity could be created, for example called Project, which would be related to the Experiments entity in a 1 to N relation.

There are also many improvements possible in the tables and graphs generation. One of them would be add a new section the profile of the researcher that was a dashboard, in which we were able to create and manage own graphs, combining several elements, or types. Another improvement would be the inclusion of a favourite button in the charts and table so that they could appear automatically in the dashboard.

All this changes involves two new entities in the back-end model, such as Dashboard, which connects researchers with graphs, and Chart, which saves its own information and configuration.